

q-bio 2018

Summer School

Matthew Fricke

Research Assistant Professor Applications Scientist



Biological Computation Lab



Models

A model only makes sense in terms of some relation that is preserved.

- A model that makes *predictions* about some system
- A models used to define *computation*
- *Existence proof* models (models demonstrating the possibility of something).
- A model used to *explain* something that already happened.



Models

- `Now it would be very remarkable if any system existing in the real world could be *exactly* represented by any simple model. However, cunningly chosen parsimonious models often do provide remarkably useful approximations.`
- `For such a model there is no need to ask the question "Is the model true?". If "truth" is to be the "whole truth" the answer must be "No". The only question of interest is "Is the model illuminating and useful?"`.

Box, G. E. P. (1979), "Robustness in the strategy of scientific model building", in Launer, R. L.; Wilkinson, G. N., Robustness in Statistics, Academic Press, pp. 201–236.

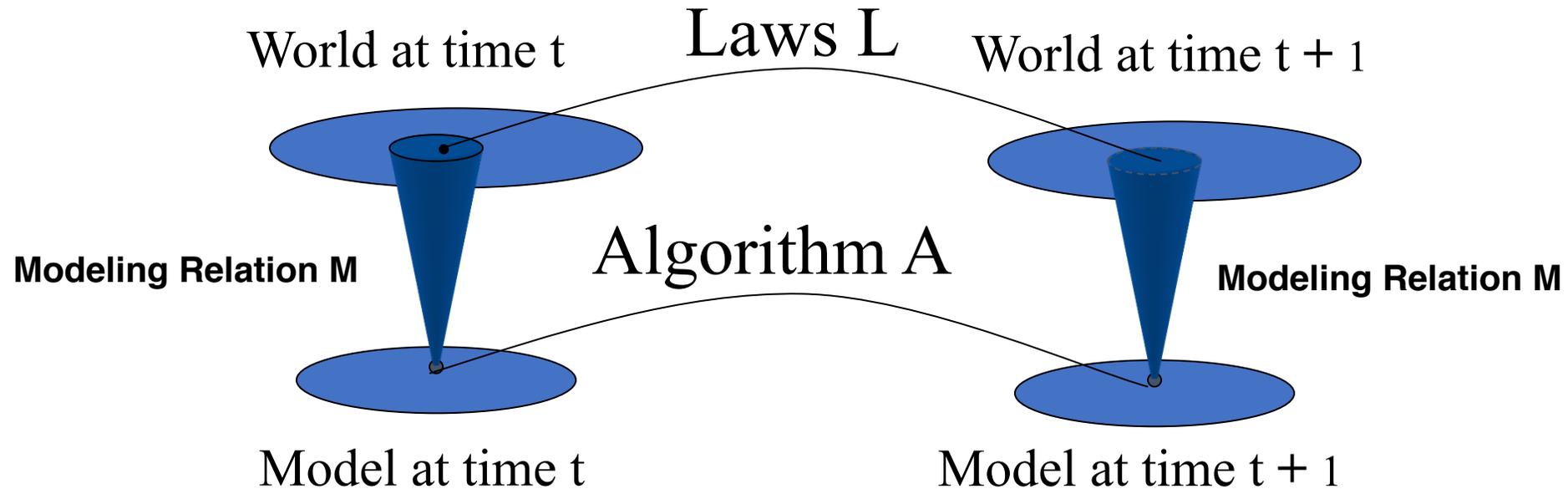
Models

- `Now it would be very remarkable if any system existing in the real world could be *exactly* represented by any simple model. However, cunningly chosen parsimonious models often do provide remarkably useful approximations.`
- `For such a model there is no need to ask the question "Is the model true?". If "truth" is to be the "whole truth" the answer must be "No". The only question of interest is "Is the model illuminating and useful?"`.
- “All models are wrong, some are useful.”

Box, George. E. P. (1979), "Robustness in the strategy of scientific model building", in Launer, R. L.; Wilkinson, G. N., Robustness in Statistics, Academic Press, pp. 201–236.

Models as Homomorphic Maps

Commutativity of the Diagram



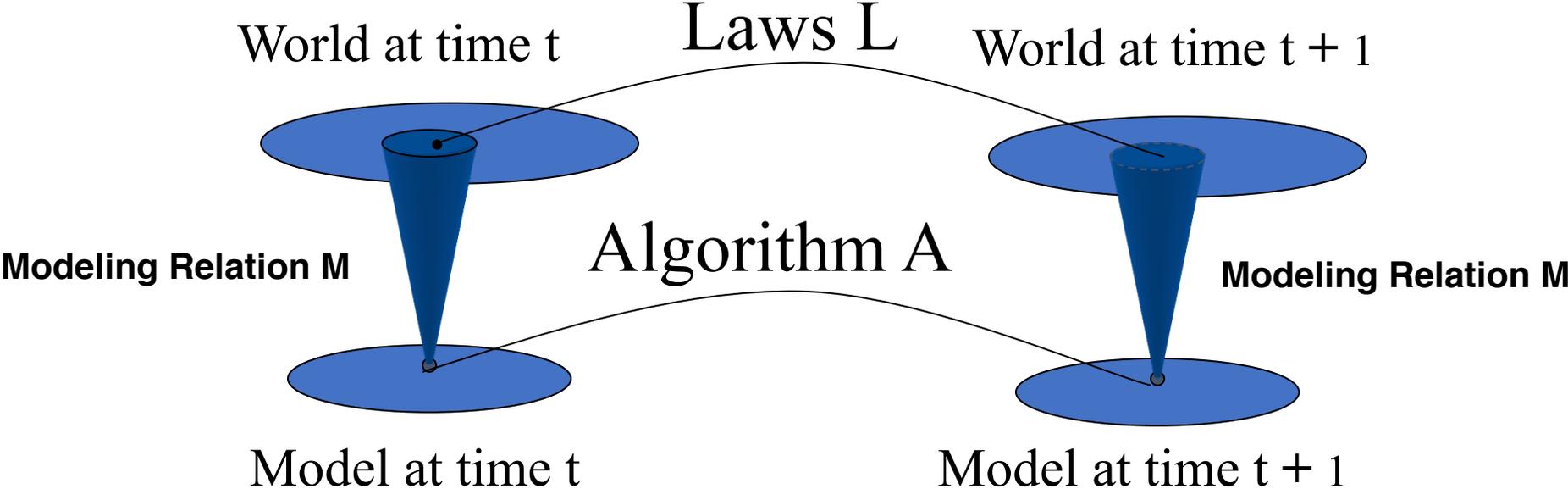
M is an equivalence relation.

Model M is valid if this is a homomorphic map:

$$M(L(x)) = A(M(x))$$

Models as Homomorphic Maps

transformation of one set into another that preserves in the second set the relations between elements of the first.



Models

- “It can scarcely be denied that the supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate representation of a single datum of experience.”

Attributed to Albert Einstein in “On the Method of Theoretical Physics,” the Herbert Spencer Lecture, Oxford, June 10, 1933. This is the Oxford University’ Press

Equivalence Classes

- Equivalence class =

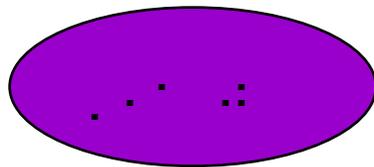
$\{x \mid x \in R\}$ and R is an equivalence relation.

- R is an equivalence relation:

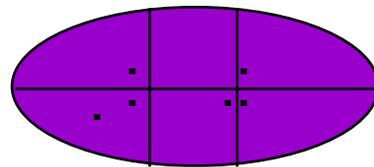
- Reflexive: $\forall x(xRx)$
- Symmetric: $xRy \Rightarrow yRx$
- Transitive: $(xRy) \wedge (yRz) \Rightarrow (xRz)$

The relation does not change unless world changes, the relation is preserved between the model and world, the model and world stay consistent over time.

- Example: $xRy \iff x$ and y are in the same little box.



Set of Objects



Partition set into 6 little boxes



Equivalence classes

Examples of Equivalence Relations

- “Is similar to” or “congruent to” on the set of all triangles.
- Logical equivalence of statements in logic.
- “Has the same image under a function” on the elements of the domain of the function.
- What’s not an equivalence relation?
 - The relation “ \geq ” between real numbers is reflexive and transitive, but not symmetric. For example, $7 \geq 5$ does not imply that $5 \geq 7$. It is, however, a partial order.
 - The relation “is a sibling of” on the set of all human beings is not an equivalence relation.
 - Is Symmetric (if A is a sibling of B, then B is a sibling of A)
 - Not reflexive (no one is a sibling of himself),
 - Not transitive (since if A is a sibling of B, then B is a sibling of A, but A is not a sibling of A).

Example Homomorphism:

Multiplication of Integers

- Model all pairs of integers and their product:
 - e.g., $14792 \times 4183584 = 61883574528$
- Model:
 - Even X Even = Even
 - Even X Odd = Even
 - Odd X Even = Even
 - Odd X Odd = Odd

Example Homomorphism:

Multiplication of Integers

Model:

Even x Even = Even

Even x Odd = Even

Odd x Odd = Odd

Odd x Even = Even

$$M(L(x)) = M(2^n \times 2^m = 2^k) = \text{Even} \times \text{Even} = \text{Even}$$

The relationships are preserved under our model.

Model relationship:

$$2^n \times 2^m = 2^k \ R \ \text{Even} \times \text{Even} = \text{Even}$$

$$2^{n+1} \times 2^{m+1} = 2^{k+1} \ R \ \text{Odd} \times \text{Odd} = \text{Odd}$$

$$2^n \times 2^{m+1} = 2^k \ R \ \text{Even} \times \text{Odd} = \text{Even}$$

$$2^{n+1} \times 2^m = 2^{k+1} \ R \ \text{Odd} \times \text{Even} = \text{Even}$$

Lattice Gas Models (LGCA)

- Gasses and fluids can be modelled with continuous models
- That is, we can use continuous values of pressure, temperature, and velocity

Lattice Gas Models

- Gasses and fluids can be modelled with continuous models
- That is, we can use continuous values of pressure, temperature, and velocity
- What happens when we get to extreme cases:
 - If we are modelling a disk drive head moving just a micron above the platter these continuous models break down.
 - We have to model the individual molecules of gas.

Lattice Gas Models

- Gasses and fluids can be modelled with continuous models
- That is, we can use continuous values of pressure, temperature, and velocity
- What happens when we get to extreme cases:
 - If we are modelling a disk drive head moving just a micron above the platter these continuous models break down.
 - We have to model the individual molecules of gas.
- If we are modelling systems with very high energies (such as a nuclear explosion) we have to have a discrete model of the internal states of the atoms involved.

Lattice Gas Models

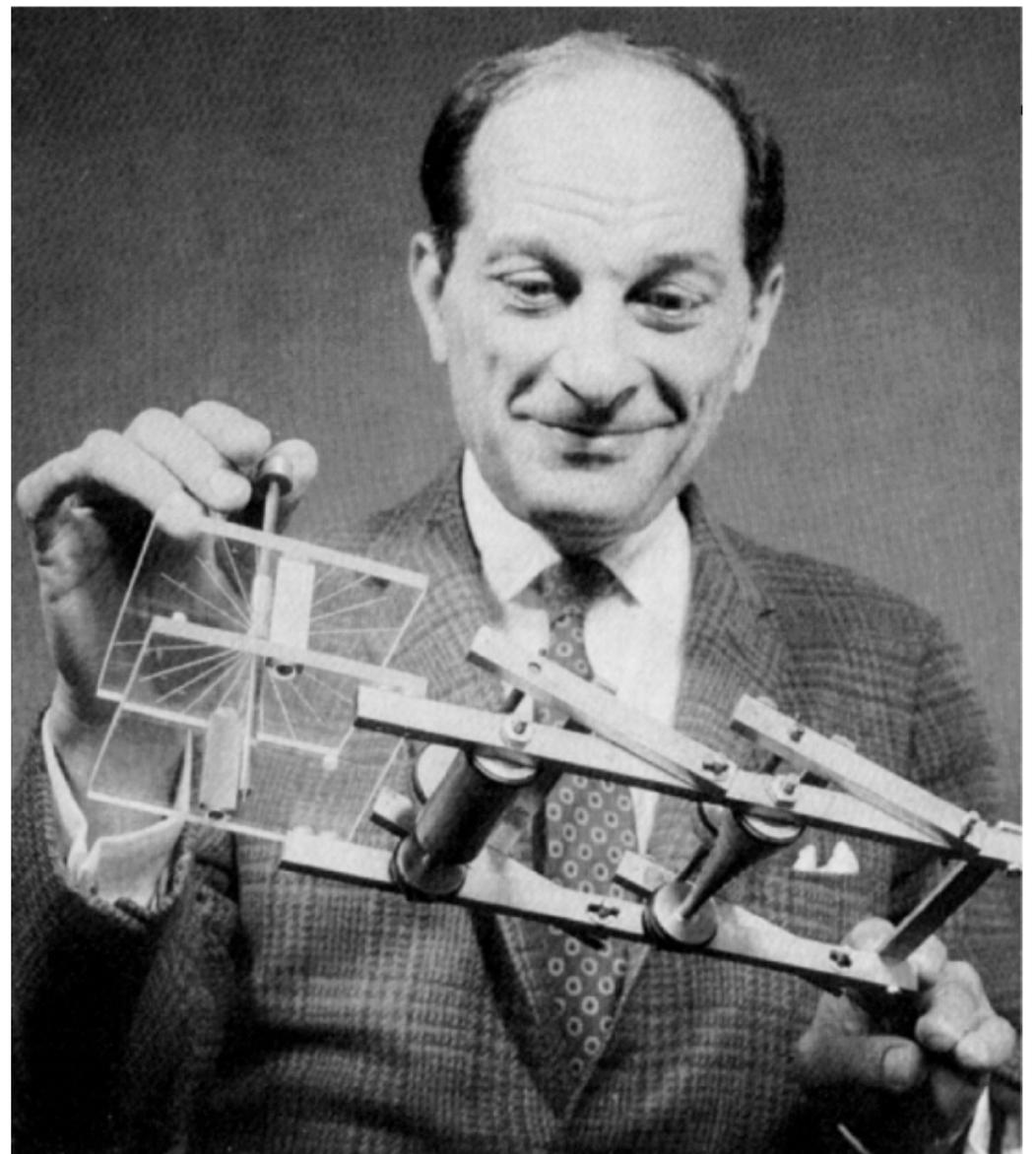
- If the molecules are very cold quantum effects start to dominate their interactions.
 - Here we have to model the quantum effects explicitly.

Lattice Gas Models

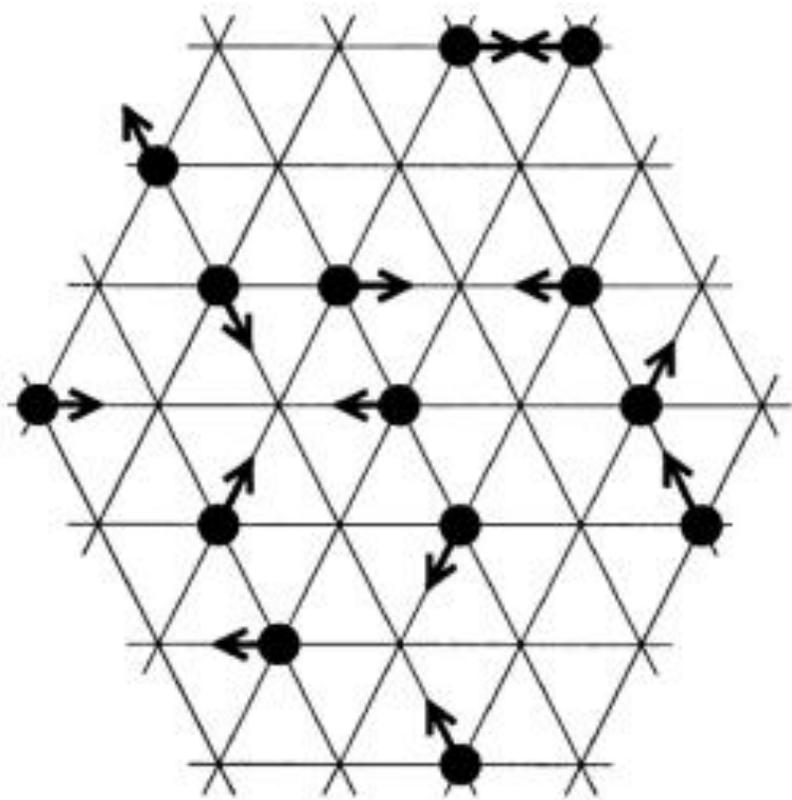
- If the molecules are very cold quantum effects start to dominate their interactions.
 - Here we have to model the quantum effects explicitly.
- These systems require models of the **microscopic** behaviour
- Models that are able to describe the behaviour of the system using just pressure, velocity, and temperature are **macroscopic**.
- Of course, we could model all gasses and fluids at the microscopic level.

Lattice Gas Models

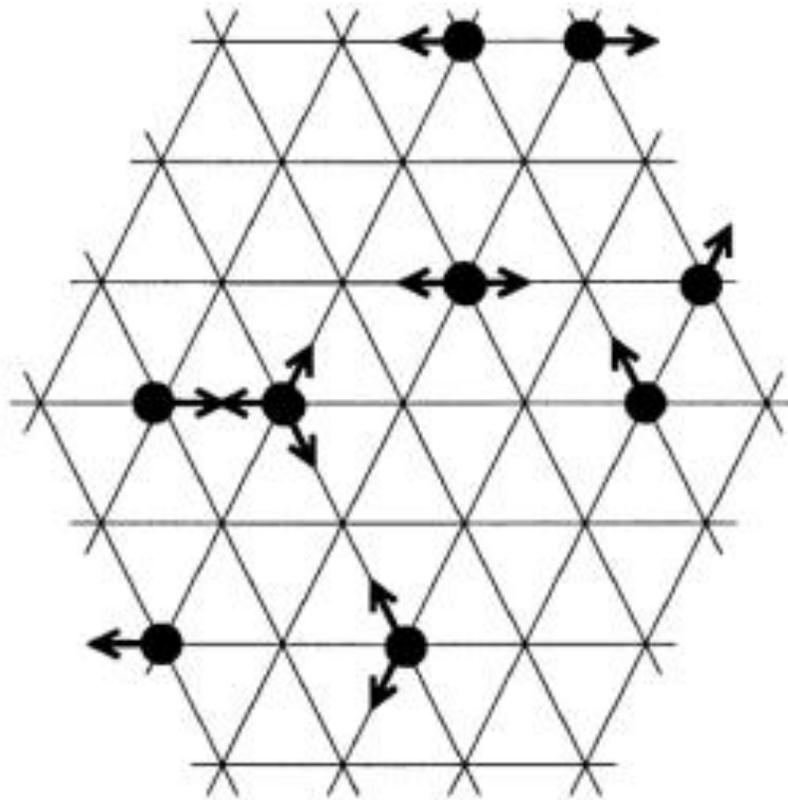
- Cellular automata are used to model molecular systems.
- The use of cellular automata to model particles such as gasses, fluids, and
- The propagation of subatomic particles was pioneered by Stanislaw Ulam and John von Neumann in the 1950s.



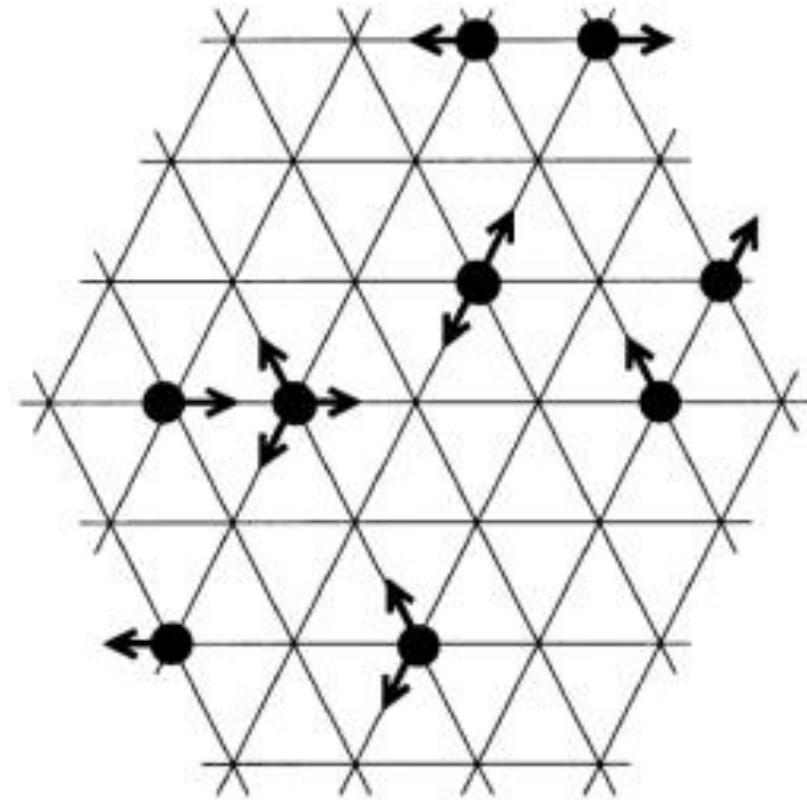
Stanislaw Ulam with the FERMIAC, used to model neutron transport, Los Alamos National Labs



(a) initial lattice

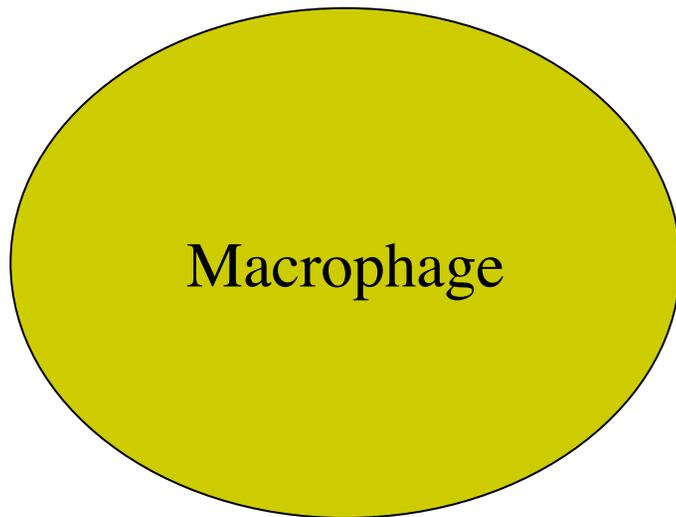


(b) propagation

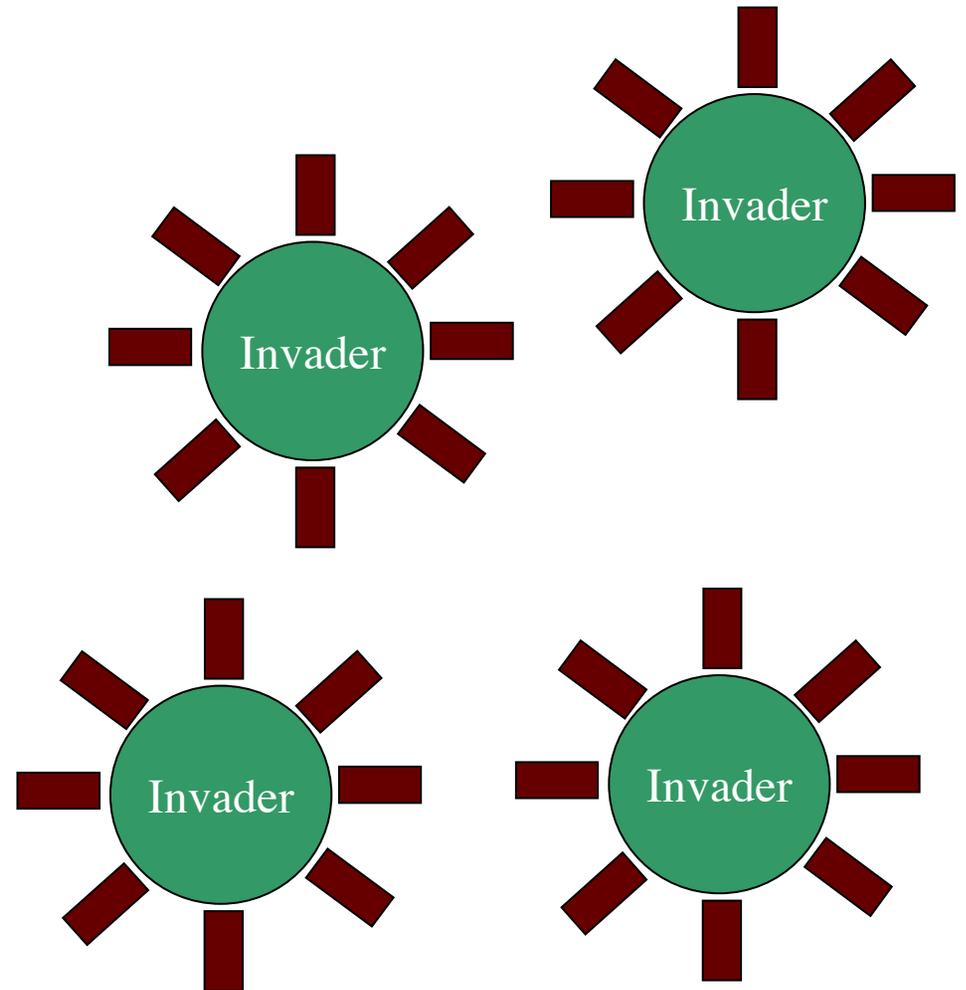


(c) collision handling

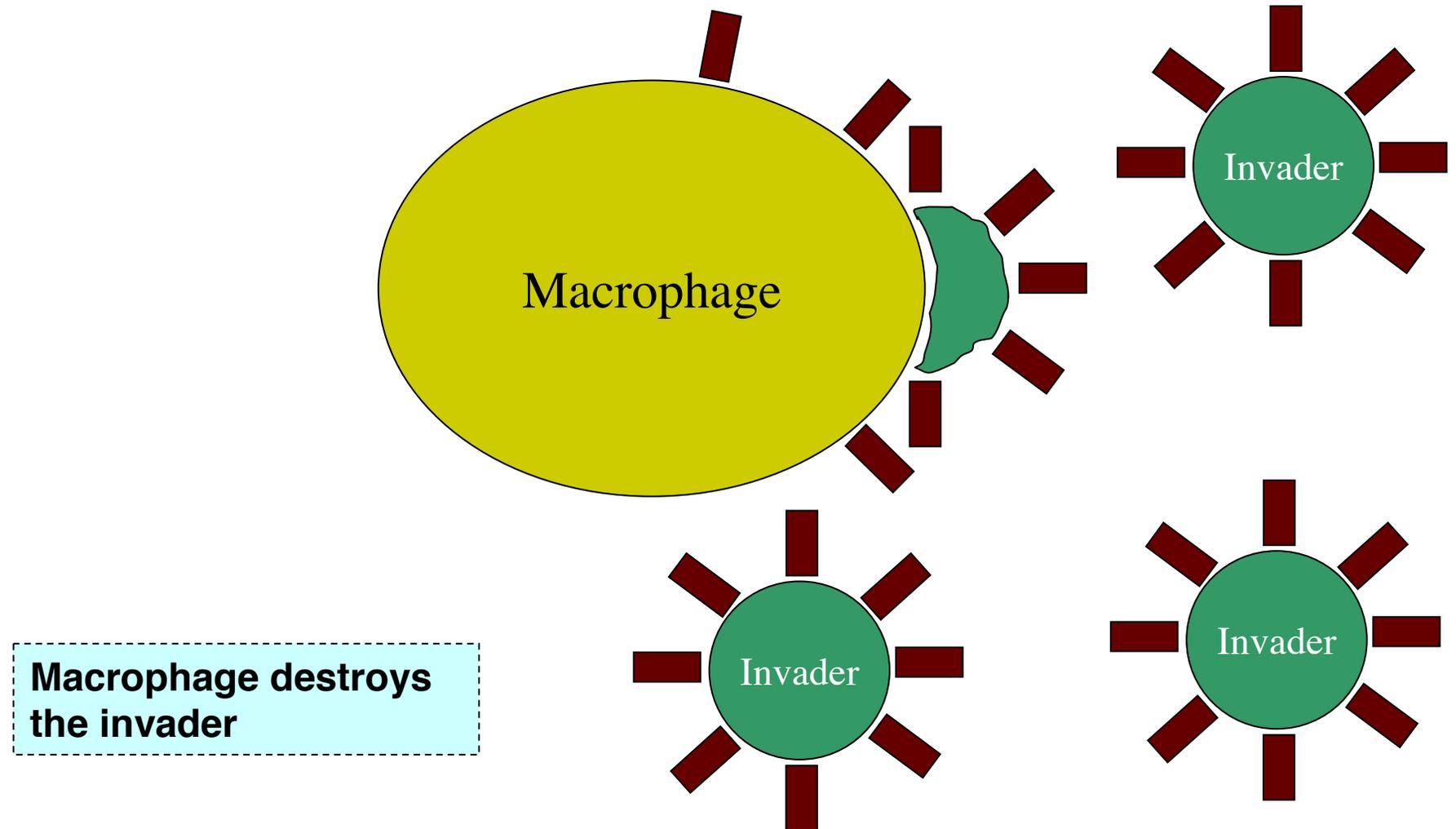
Immune System Signaling (abridged)



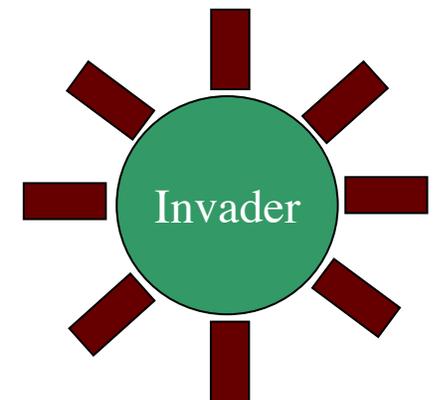
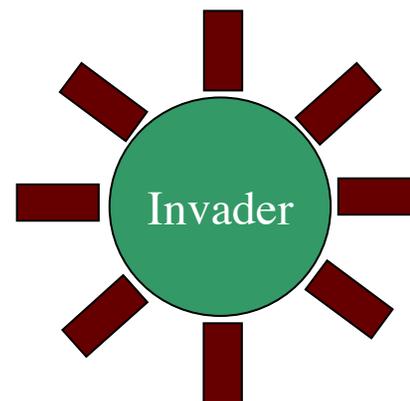
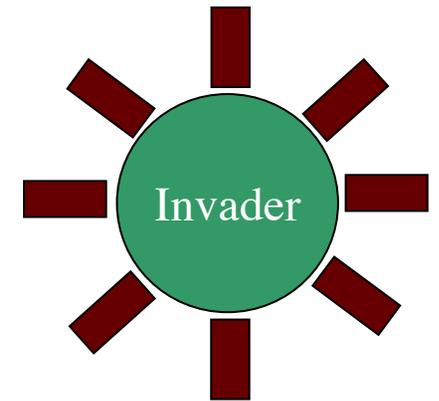
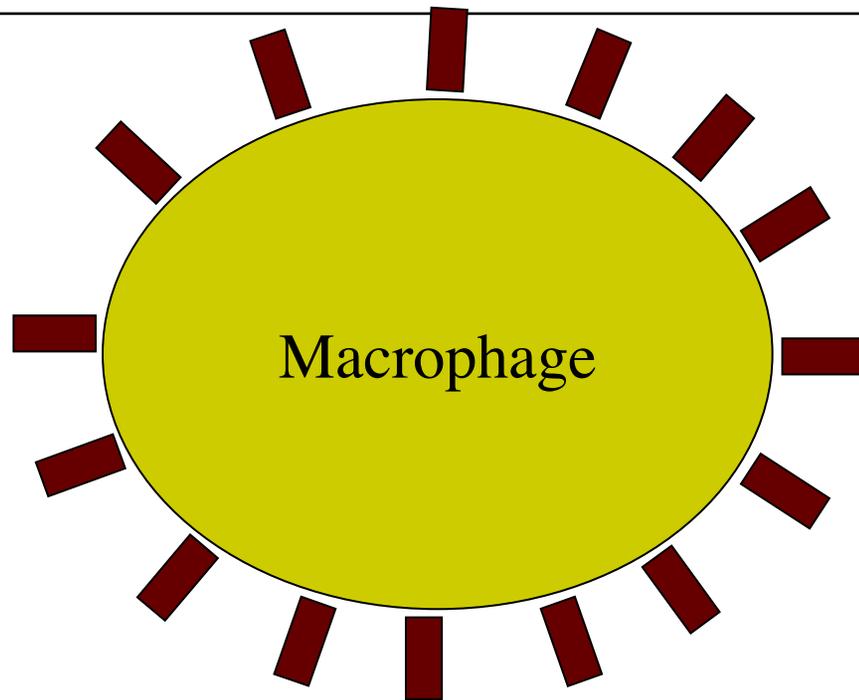
**Macrophage encounters
an invader**



Immune System Signaling (abridged)

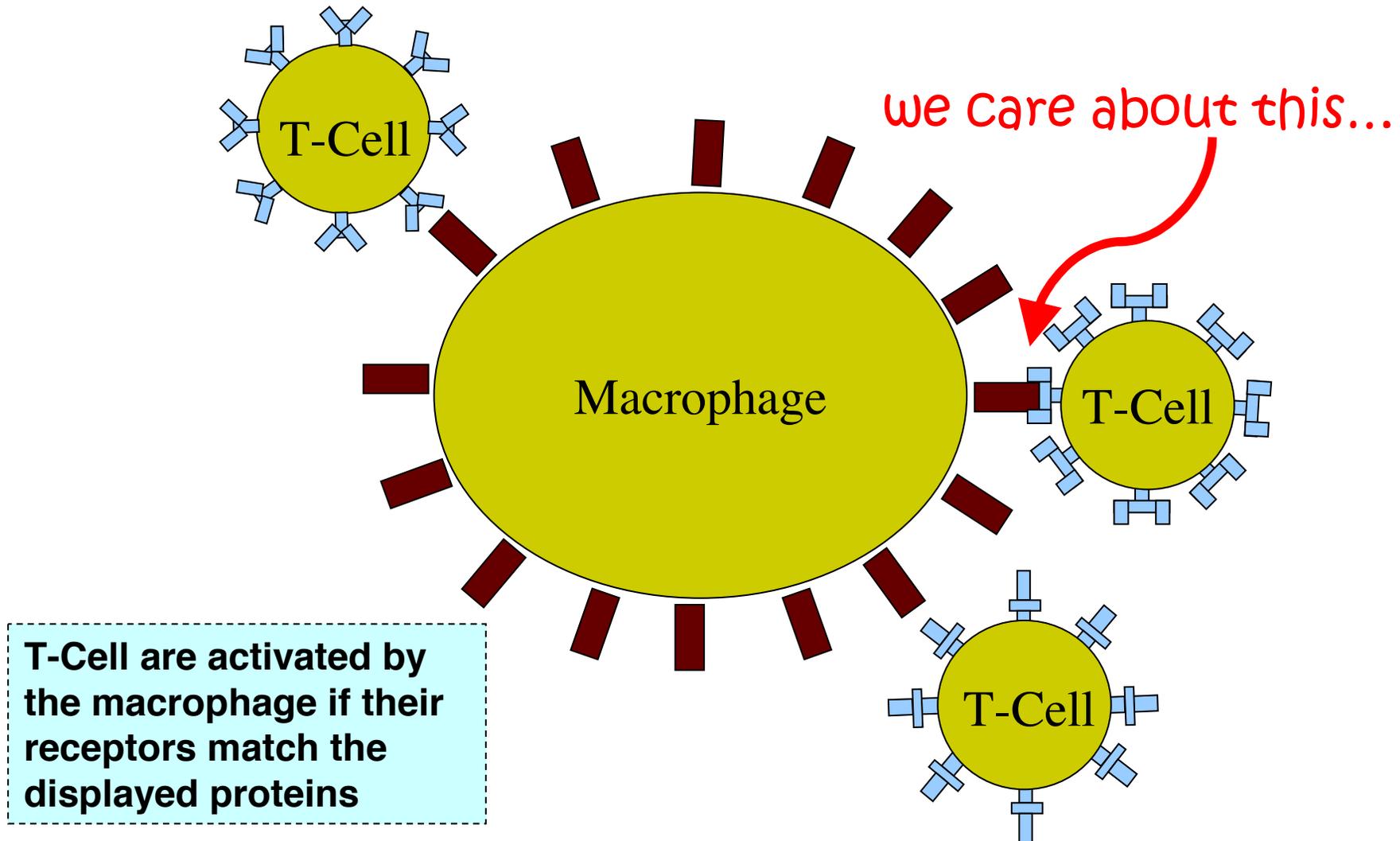


Immune System Signaling (abridged)

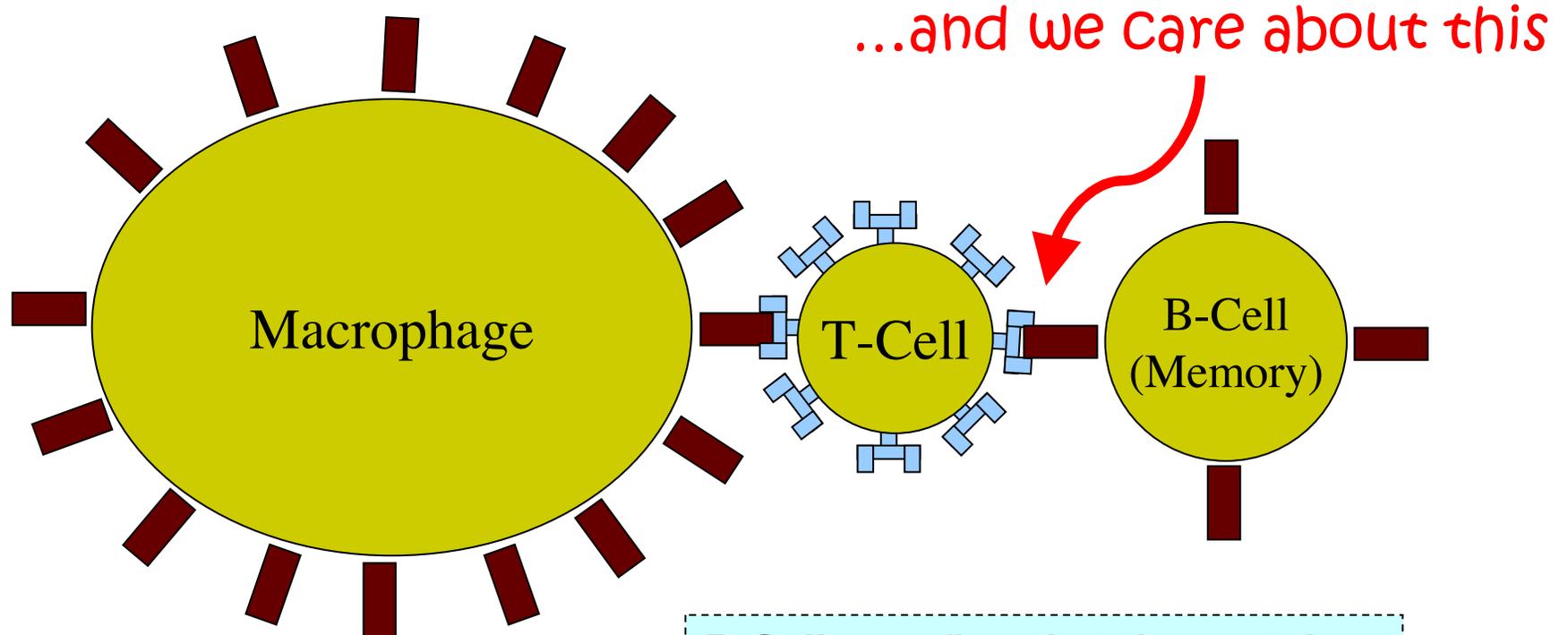


**Macrophage displays
proteins from the invader**

Immune System Signaling (abridged)



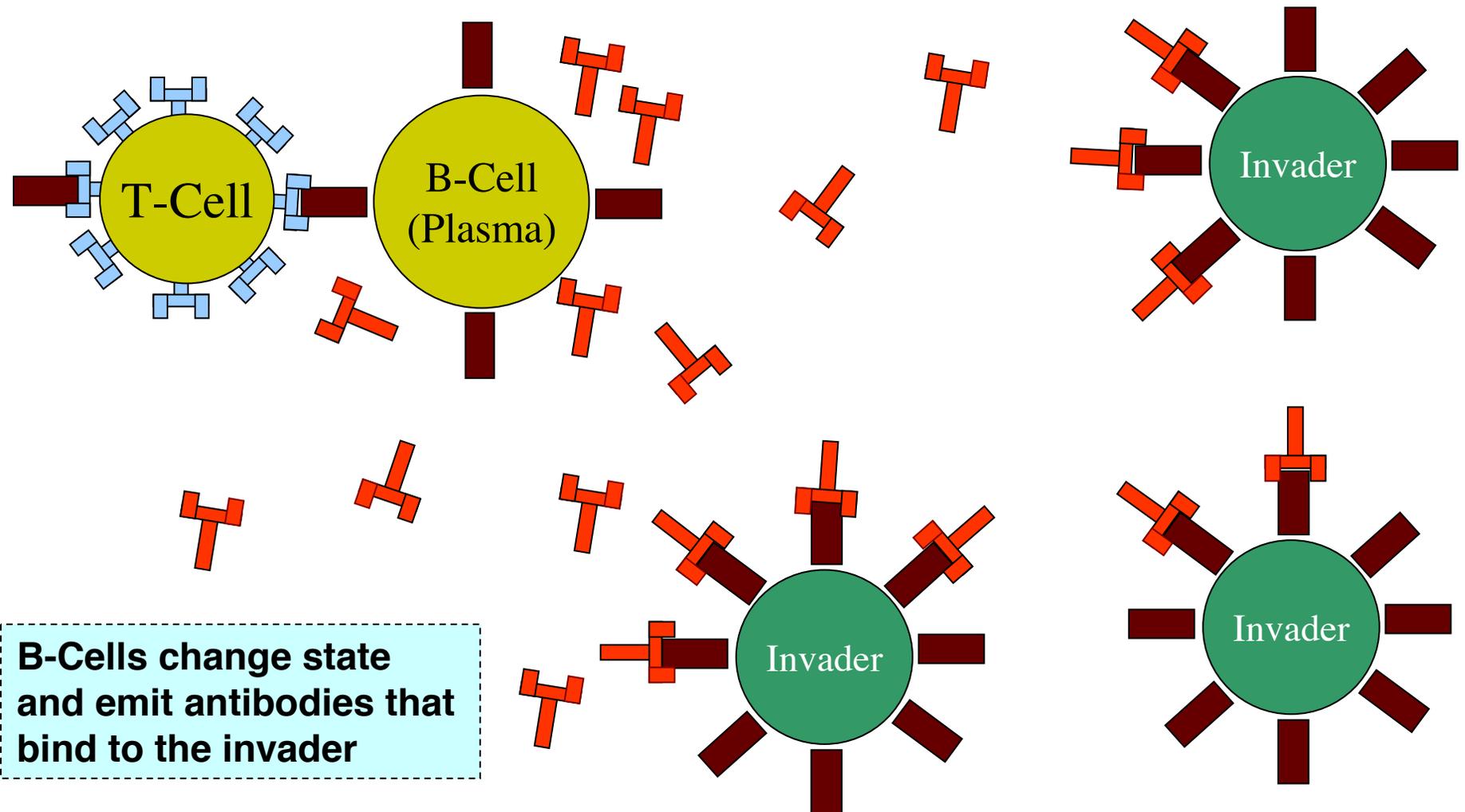
Immune System Signaling (abridged)



...and we care about this

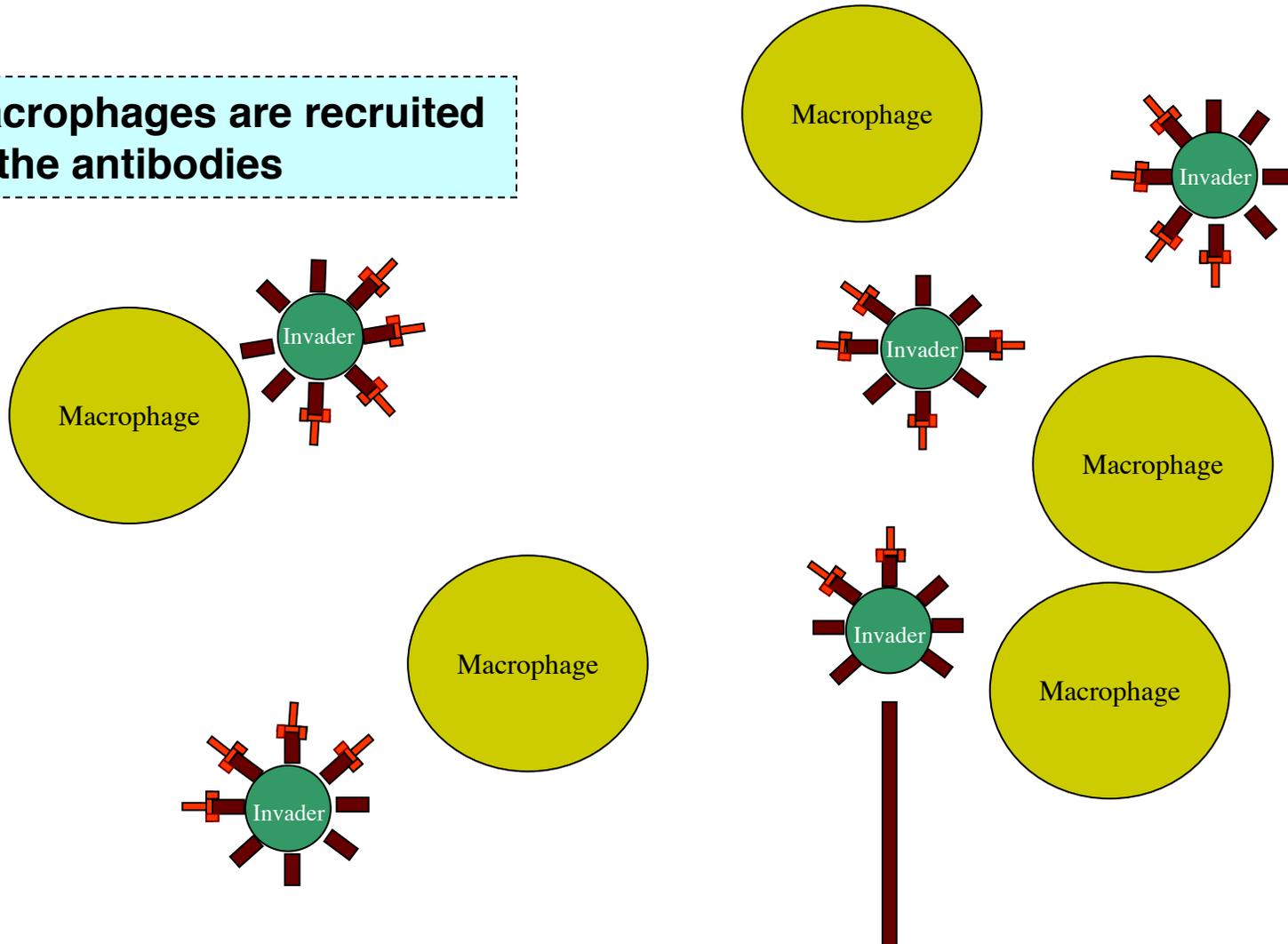
B-Cells confirm that the proteins are from a pathogen encountered previously.

Immune System Signaling (abridged)



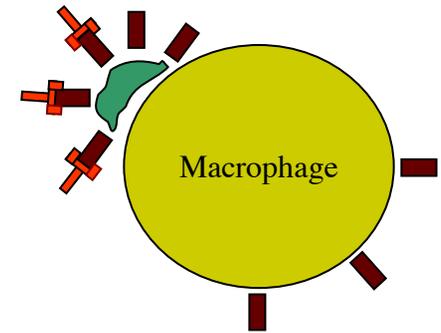
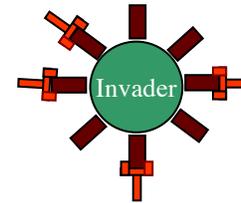
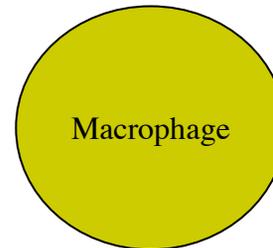
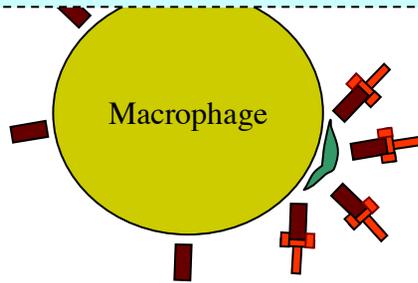
Immune System Signaling (abridged)

Macrophages are recruited to the antibodies

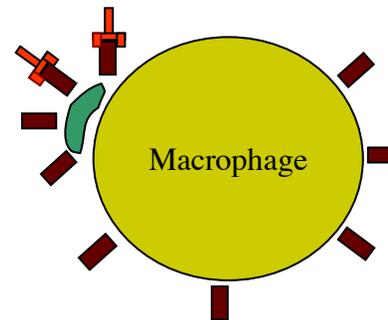
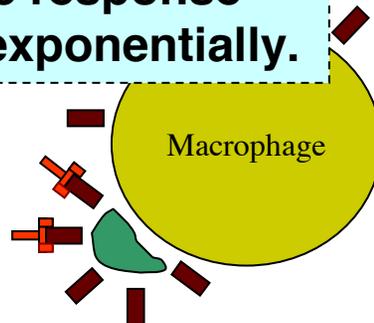


Immune System Signaling (abridged)

**Recruited macrophages
destroy the invaders and
display their proteins.**

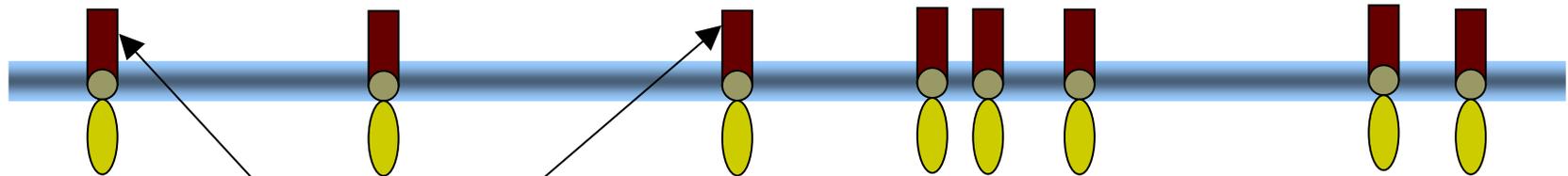
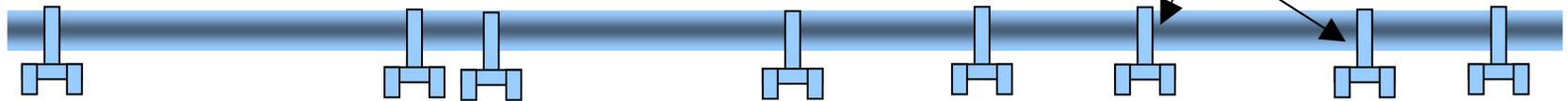


**The cycle repeats and
the immune response
increases exponentially.**



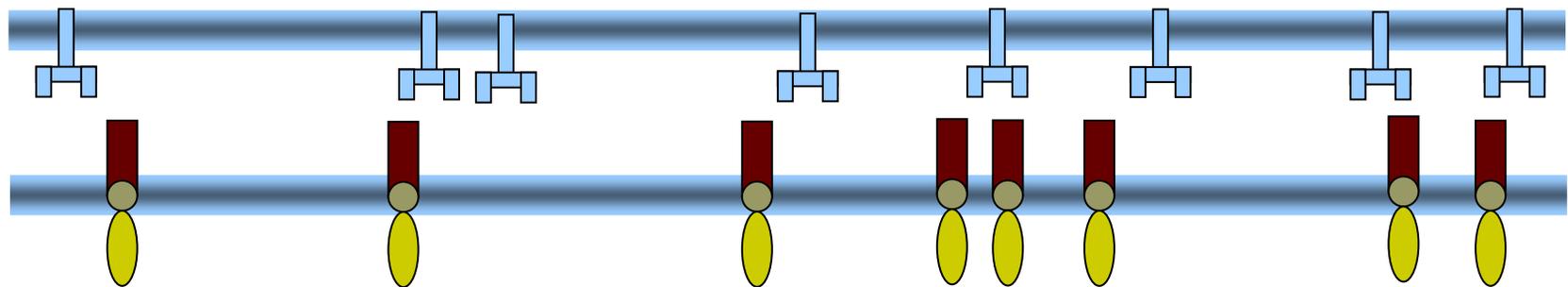
In vivo, T cells are stimulated by monovalent binding to ligands on “antigen presenting cells” (e.g. Macrophages and B-Cells)

Receptors that match particular protein snippets are displayed by T-cells.



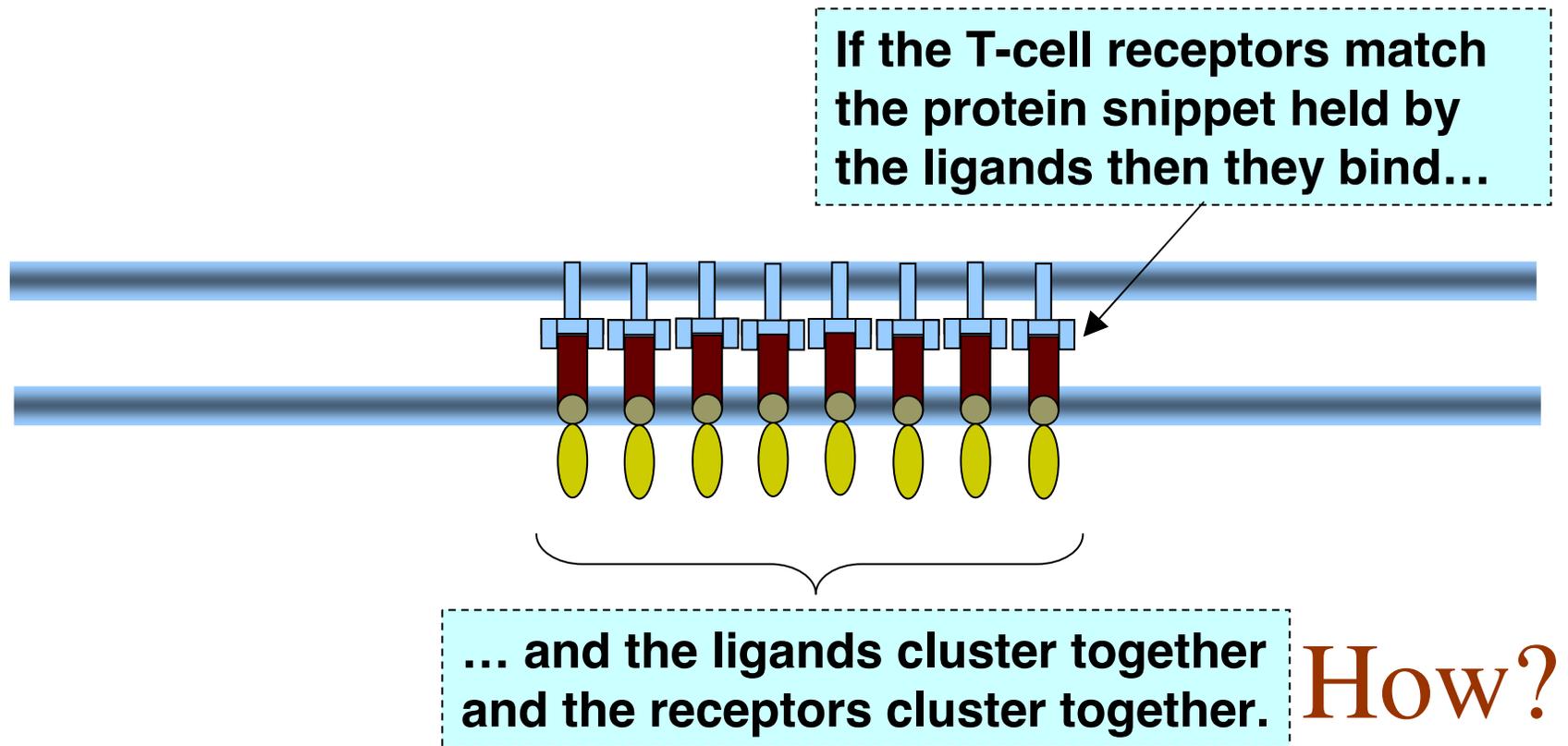
Foreign protein snippets are displayed by macrophage

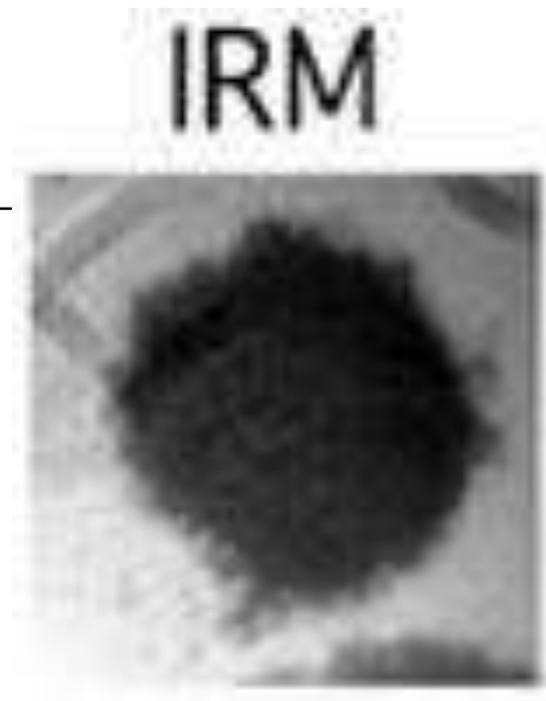
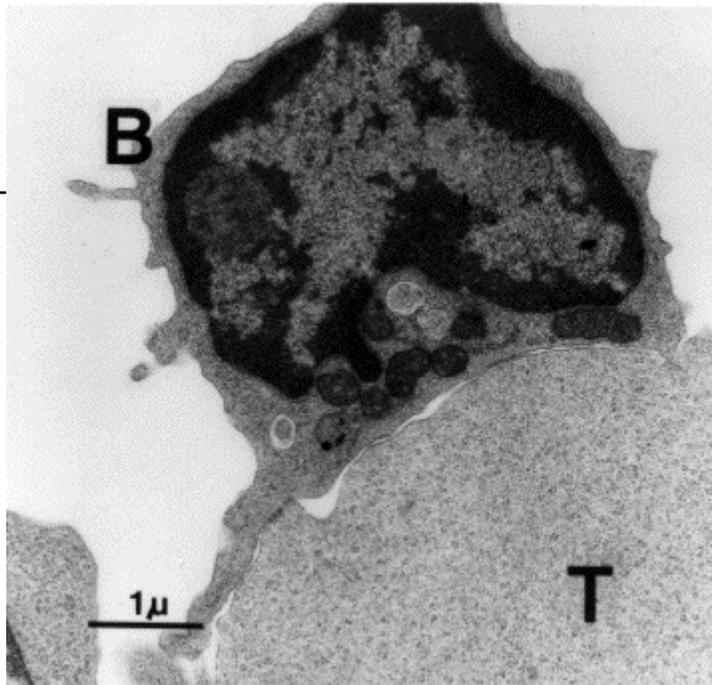
In vivo, T cells are stimulated by monovalent binding to ligands on “antigen presenting cells” (e.g. Macrophages and B-Cells)



The macrophage comes into contact with various T-cells.

In vivo, T cells are stimulated by monovalent binding to ligands on “antigen presenting cells” (e.g. Macrophages and B-Cells)





Our hypothesis: Cross Membrane binding between ligand-receptor pairs serves to combine the attractive forces between proteins in their own membranes. This would allow receptor or ligand groups that by themselves are do not cluster to “sum” the attractive forces and cluster.

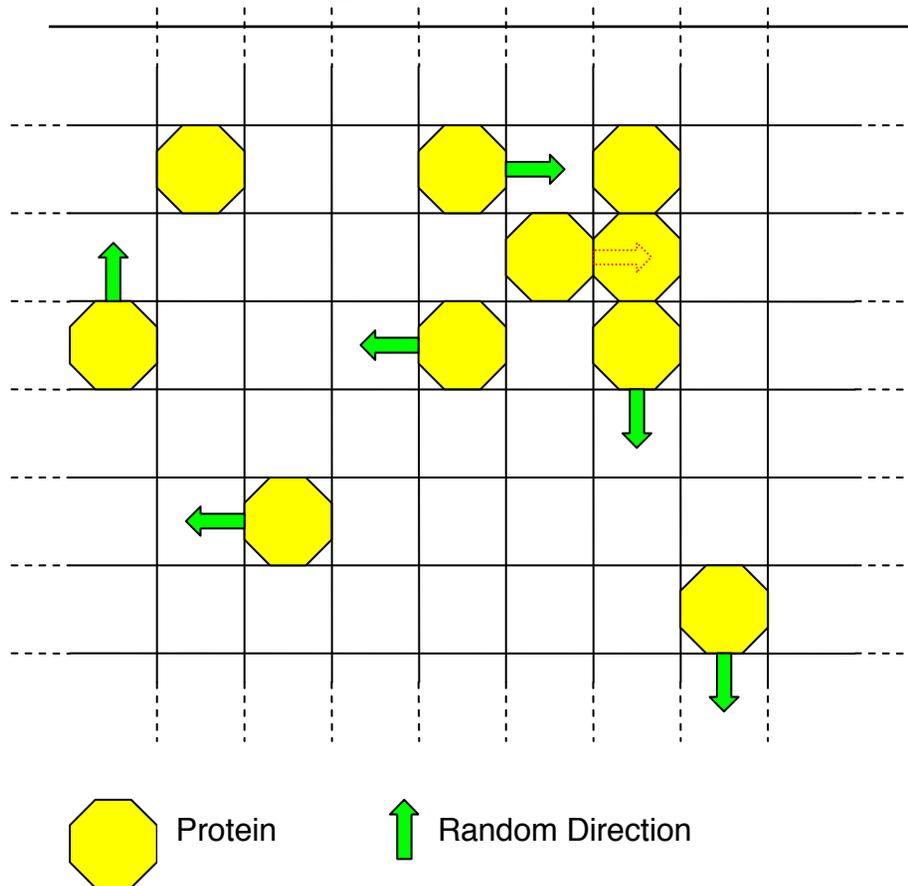
Sounds simple but we can't predict how strong the inter-membrane force needs to be in relation to the intra-membrane forces to cause phase separation. So model it!



Our Approach

- Write a model of phase separation on a single membrane
- Confirm that our results match those of previous phase transition models
- Implement two copies of the single membrane model and bring them into contact
- Add a cross-membrane binding force
- Under what circumstances do we get phase separation?

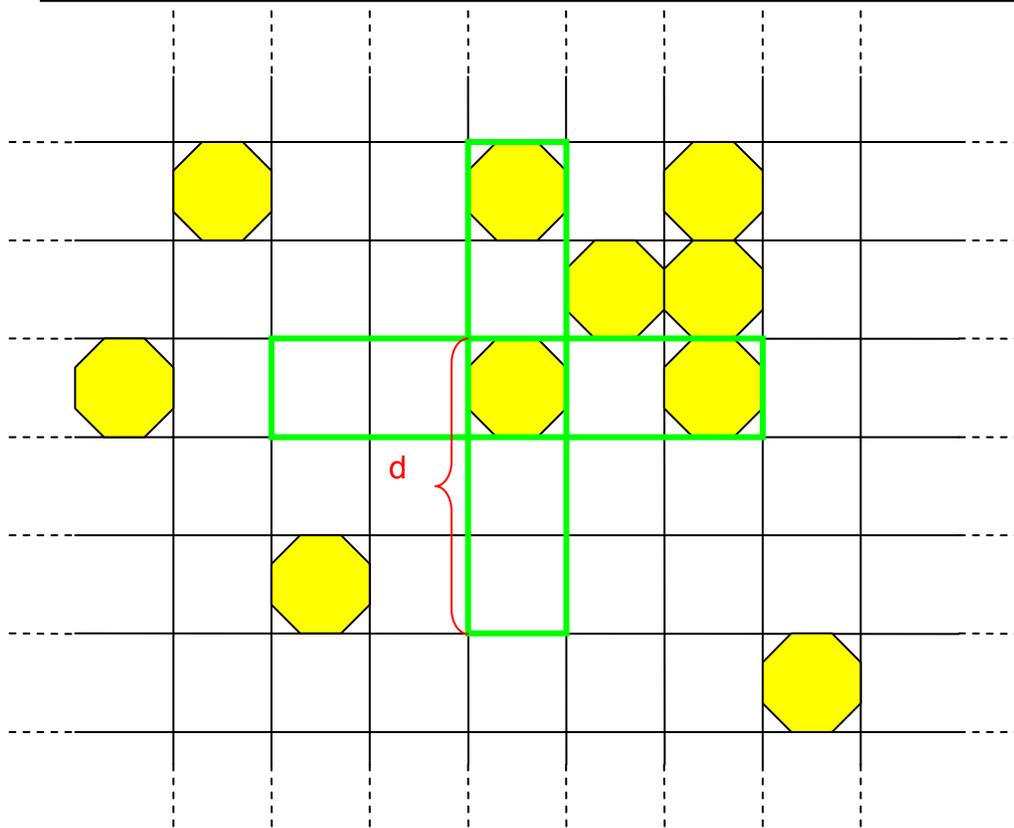
Metropolis Monte Carlo



ε = Favorable contact energy (in kT)
between neighboring proteins.

- $n \times n$ toroidal lattice
- Each site on the lattice can hold a single protein
- At each discrete time-step all proteins choose a random direction to move
- If the energy is reduced the motion is accepted.
- Otherwise the motion is accepted with probability $e^{(-\Delta E/kT)\varepsilon}$.
- Repeat until we are confident that the system is in equilibrium

Measuring Phase Separation – Spatial Autocorrelation



- Autocorrelation Function $g(d)$
- Choose a protein and count the number of proteins at distance d (then $\div 4$.)

The system probabilistically (Monte Carlo) enters a new lower energy configuration. The probability depends on how much the energy is decreased.

$$P = e^{-\Delta E/k_B T} \quad k_B \text{ Boltzmann Constant} \quad T \text{ Temperature}$$

Where the energy change is given by the binding energy N_{pp} at the proposed site, s_1 verses the current site s_0 .

$$\Delta E = - \epsilon k_B T (N_{pp,s1} - N_{pp,s0})$$

The system probabilistically (Monte Carlo) enters a new lower energy configuration. The probability depends on how much the energy is decreased.

$$P = e^{-\Delta E/k_B T} \quad k_B \text{ Boltzmann Constant} \quad T \text{ Temperature}$$

Where the energy change is given by the binding energy N_{pp} at the proposed site, s_1 versus the current site s_0 .

$$\Delta E = - \epsilon k_B T (N_{pp,s1} - N_{pp,s0})$$

Not a dynamics model! The Monte Carlo model is sampling the space of possible protein configurations. The sites S_0 and S_1 could have been chosen randomly.

The system probabilistically (Monte Carlo) enters a new lower energy configuration. The probability depends on how much the energy is decreased.

$$P = e^{-\Delta E/k_B T} \quad k_B \text{ Boltzmann Constant} \quad T \text{ Temperature}$$

Where the energy change is given by the binding energy N_{pp} at the proposed site, s_1 versus the current site s_0 .

$$\Delta E = - \epsilon k_B T (N_{pp,s1} - N_{pp,s0})$$

Not a dynamics model! The Monte Carlo model is sampling the space of possible protein configurations. The sites S_0 and S_1 could have been chosen randomly.

$$\frac{P_{i \rightarrow j}}{P_{j \rightarrow i}} = \exp \left[- \Delta E_{ij} / k_B T \right] \quad \text{Microstate reversibility: Metropolis rule.}$$

Predictions from Theory

The protein autocorrelation will scale with binding energy as:

$$\xi \propto \frac{1}{\varepsilon_c - \varepsilon}$$

Predictions from Theory

The protein autocorrelation will scale with binding energy as:

$$\xi \propto \frac{1}{\varepsilon_c - \varepsilon}$$

Time spent proteins spend bound (p) vs unbound (u):

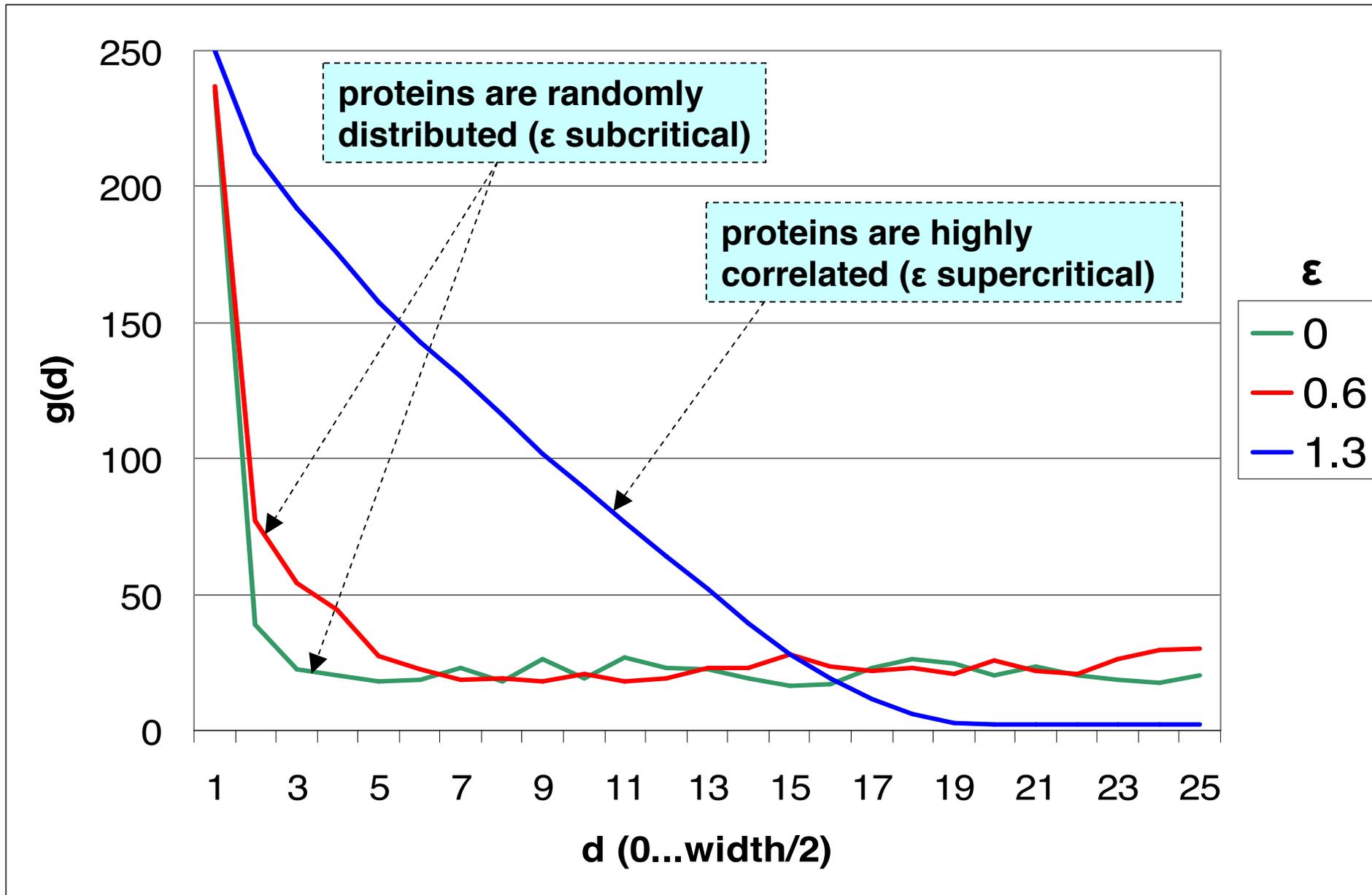
$$\frac{\overline{t_u}}{\overline{t_p}} = \frac{1 - c}{c} e^{-\varepsilon_x} \quad c \text{ is the concentration of proteins.}$$

Predictions from Theory

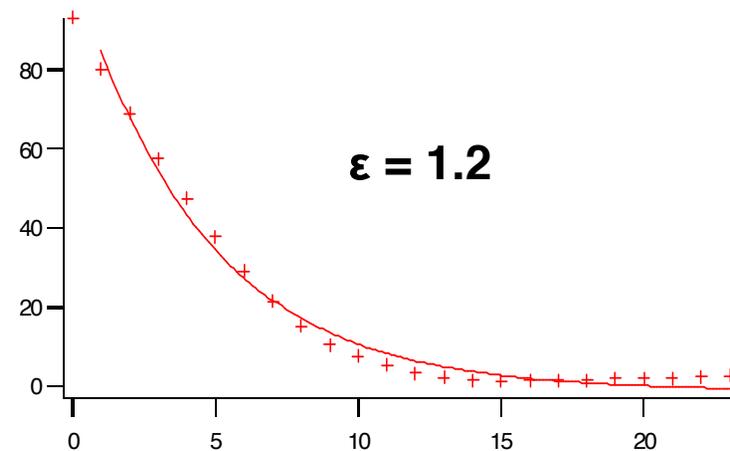
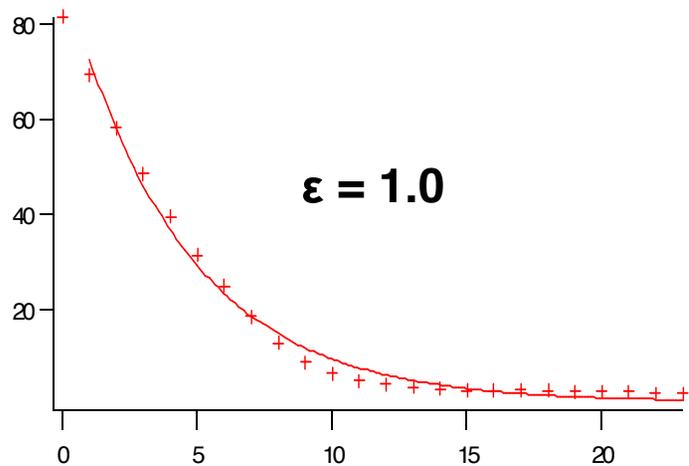
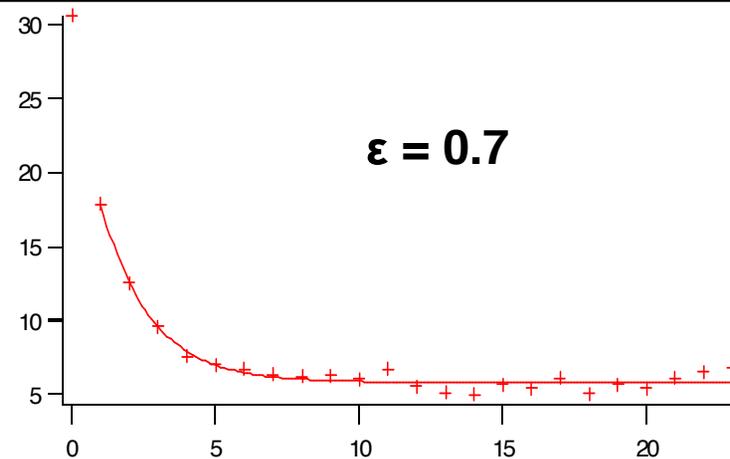
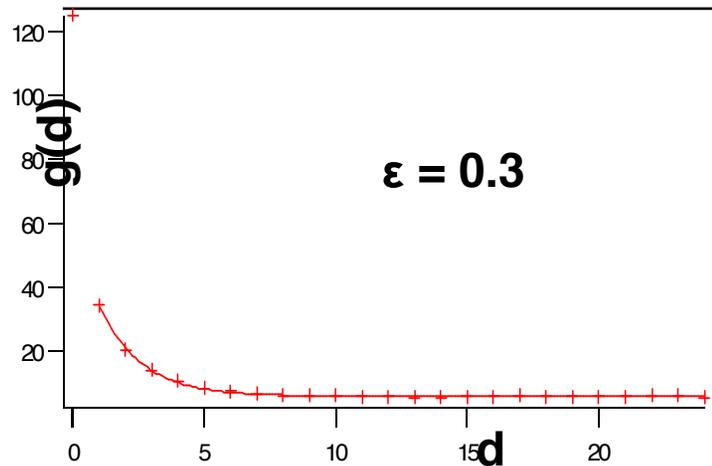
The effective interprotein interaction is:

$$\varepsilon_{\text{eff}} = \frac{\varepsilon \bar{t}_u + 2\varepsilon \bar{t}_p}{\bar{t}_u + \bar{t}_p} = \frac{(1 - c)e^{-\varepsilon_x} + 2c}{(1 - c)e^{-\varepsilon_x} + c} \varepsilon$$

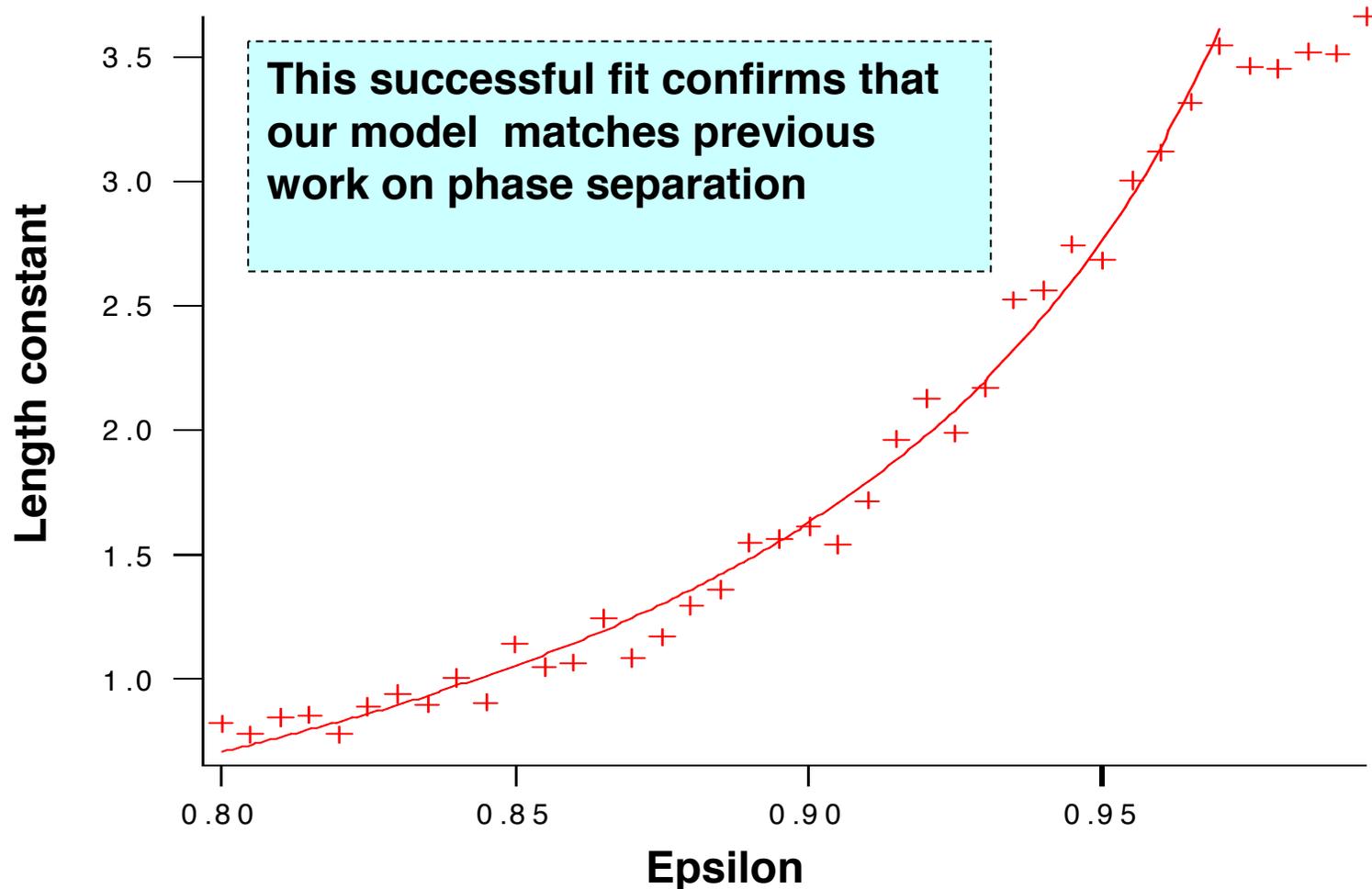
Correlation Functions for Three Values of ϵ



Fit Correlation Functions to an Exponential $y = C_1 e^{C_2 x}$ (fit deteriorates as critical epsilon reached)

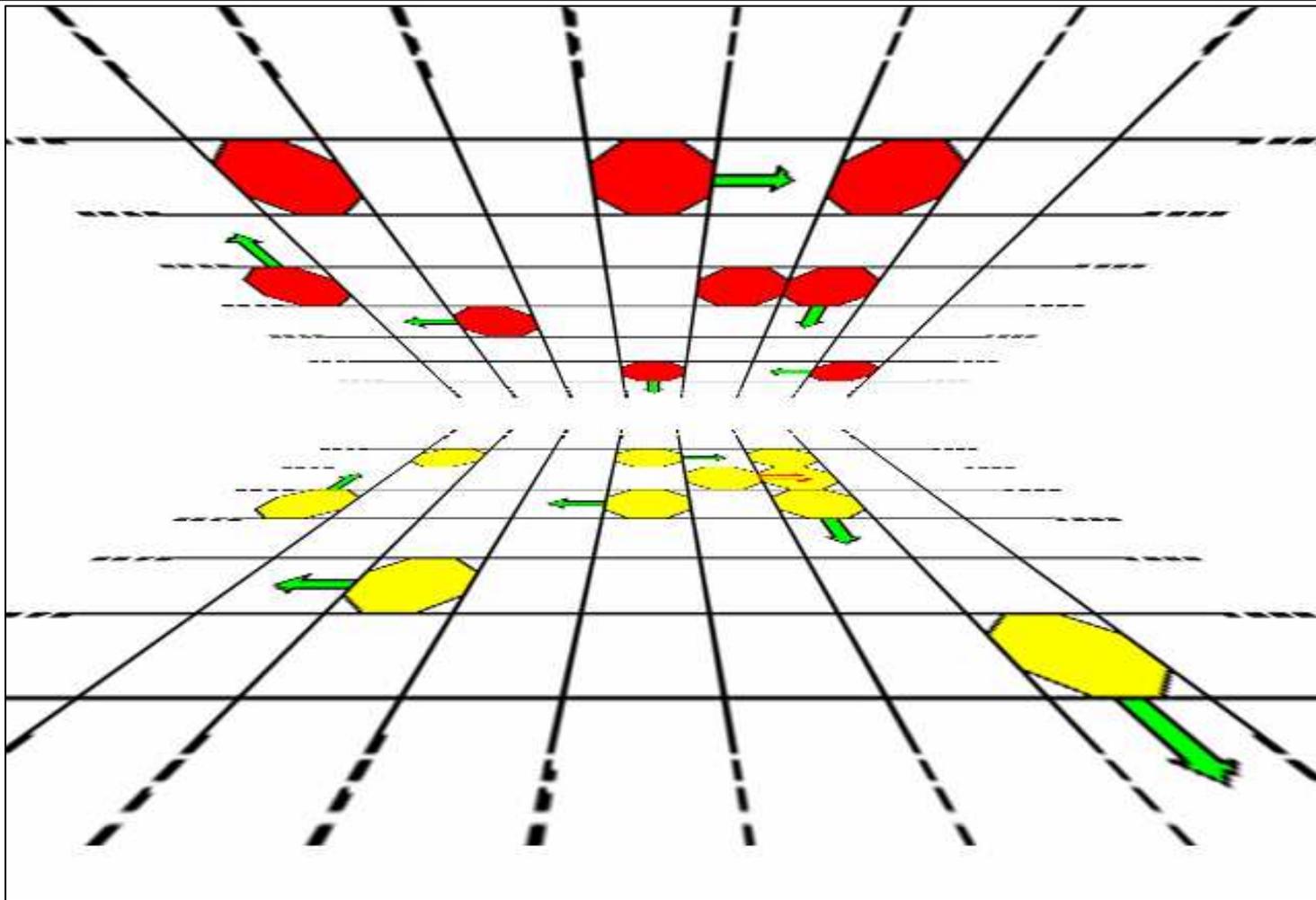


Fit exponential length constants to $\frac{C_1}{\varepsilon - \varepsilon_c} + C_2$ *



* Gould H., and J. Tobochnik An Introduction to Computer Simulation Methods: Applications to Physical Systems, 1996

Two Membrane Model

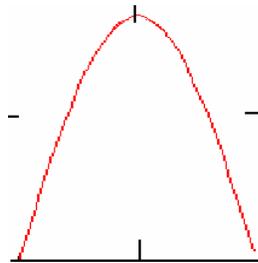
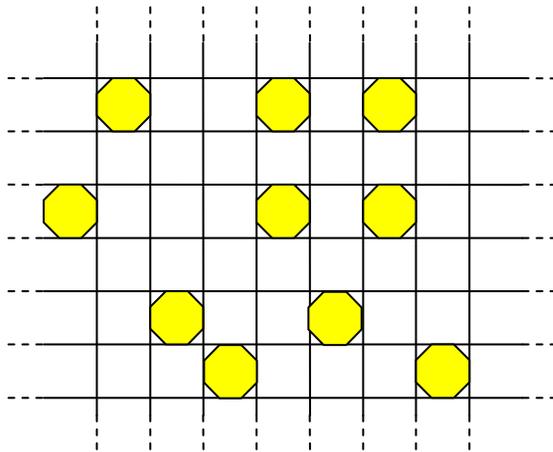


Density variance as a measure of phase separation

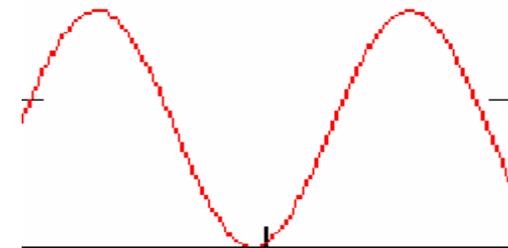
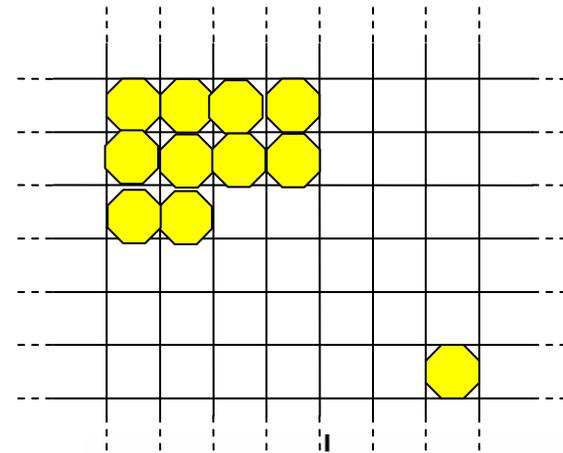
Calculating the autocorrelation, exponents, and critical exponents is too slow

Instead: calculate the protein density for all overlapping 3x3 squares on the lattice

Standard deviation is a measure of phase separation

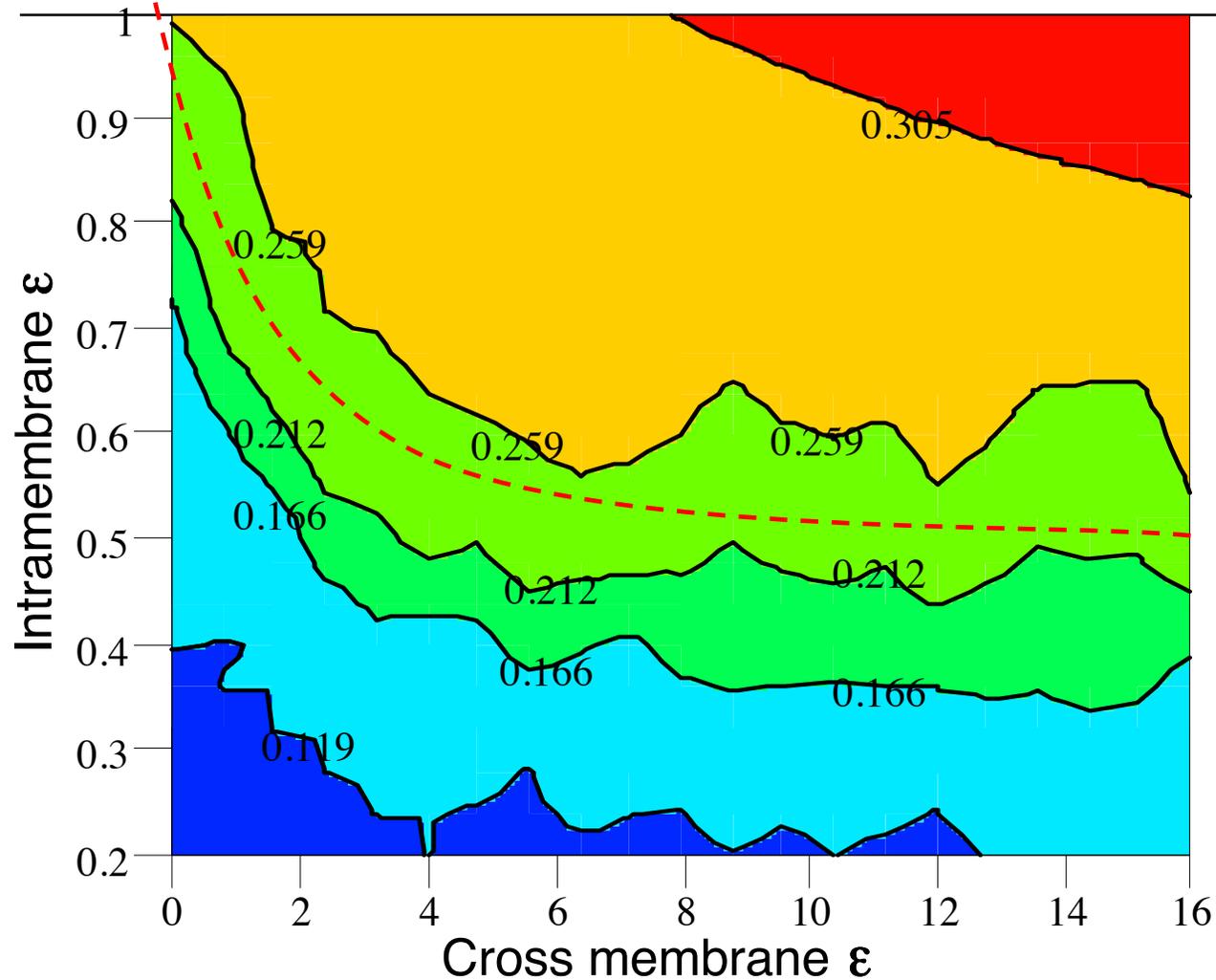


Low σ , one phase



High σ , two phase

Contour Plot of Phase Separation



Complete phase separation occurs at 0.27

Random protein distributions have been observed to have values of between 0.09 and 0.105

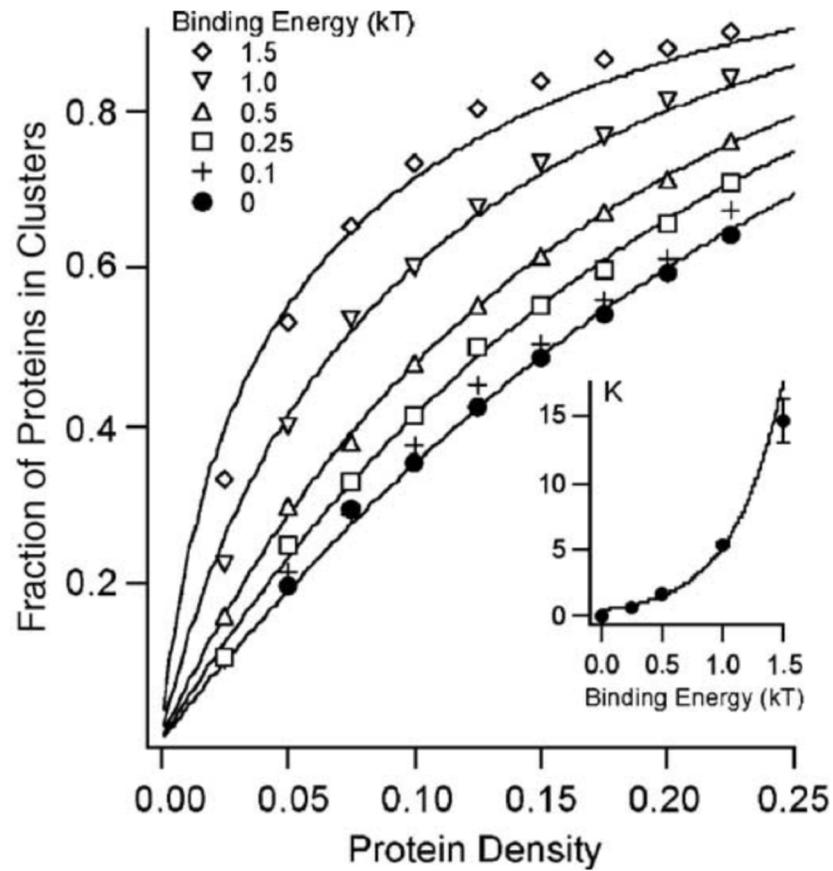


Fig. 2. Fraction of proteins in clusters (dimers or higher aggregates), as a function of protein density, for various attractive interaction energies. The clustered fraction was fit to a background level of statistical aggregation, plus a mass action term (a dimerization equilibrium). The dimerization constant K is plotted in the inset versus the binding energy, and shows the expected exponential dependence. Exact correspondence is not expected, because higher order aggregation is possible. Data from 100×100 lattice run for 10,000 iterations.

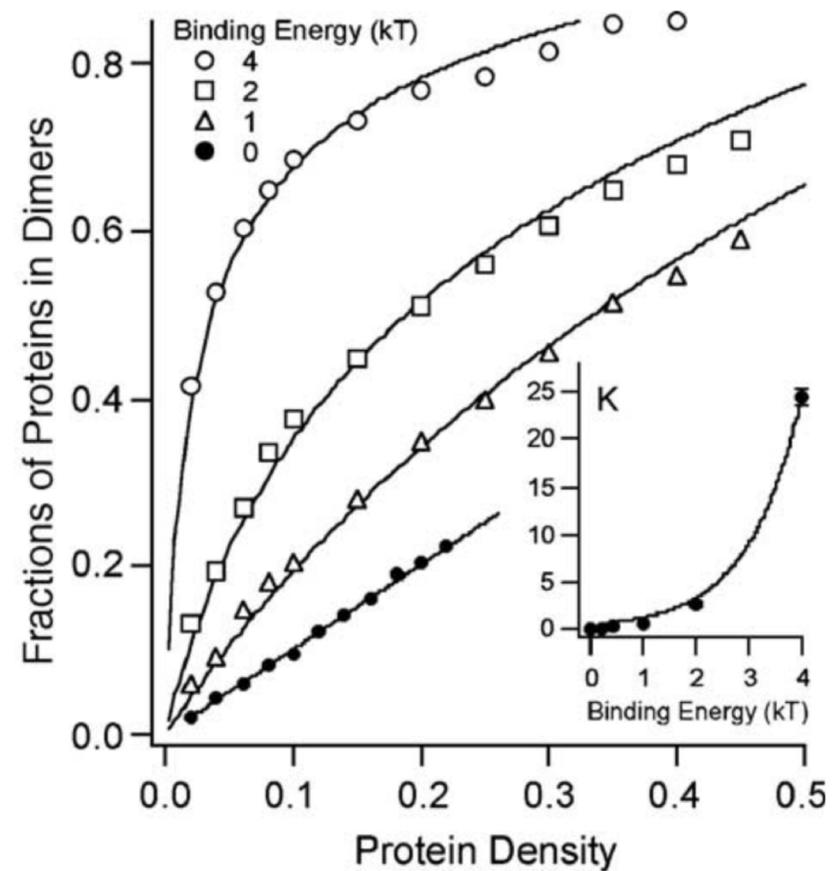


Fig. 3. Fraction of proteins in *intermembrane* dimers as a function of protein density in both membranes. At zero interaction energy, the fraction of dimers is the same as the protein density, as expected for random associations. The dimer concentration can be well fit by the sum of the background association, plus a mass action term. The dimerization constant for the mass action term is well fit to an exponential in binding energy, inset.

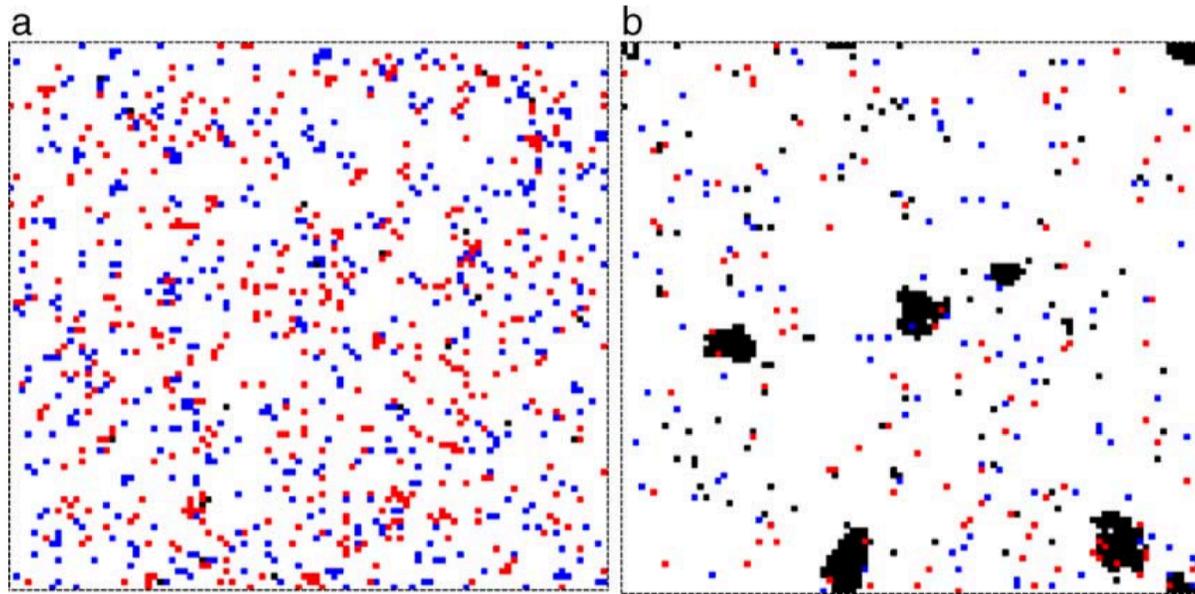


Fig. 4. (a) left: 100×100 double lattice at step 6500 with a protein concentration of 0.05 on each lattice. Intra-lattice protein interaction energy is $0.6 k_B T$ and the intermembrane interaction energy is zero. At this interaction energy, large clusters are never observed. (b) right: The same lattice configuration as in (a) but with an added intermembrane interaction energy of $5.0 k_B T$. Note the formation of large clusters as a consequence of the added intermembrane interaction.

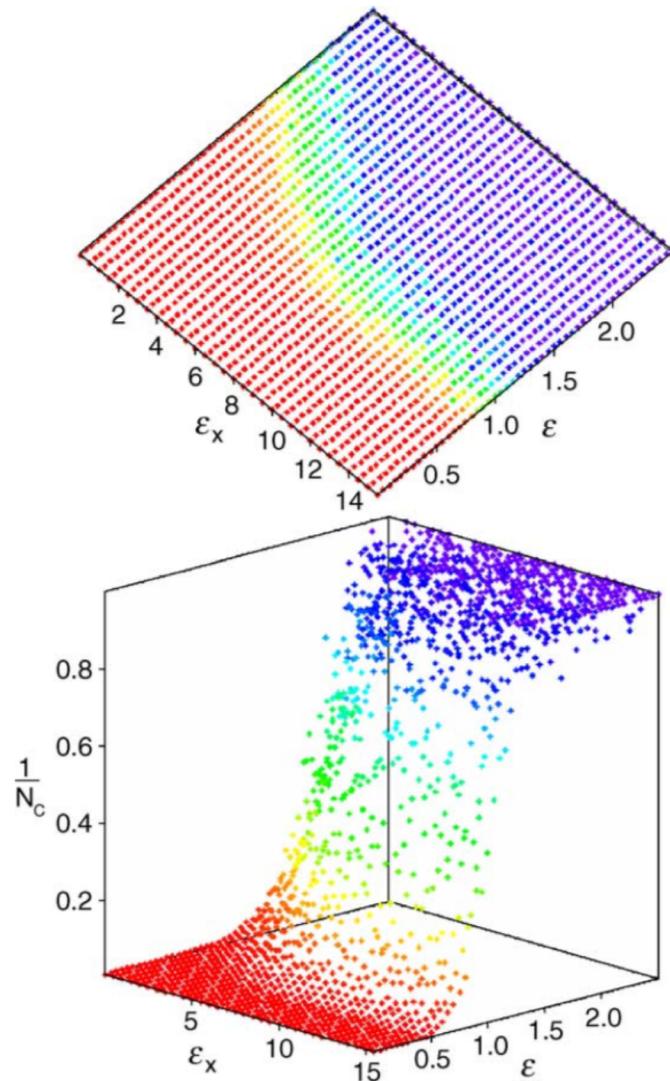


Fig. 5. The reciprocal of the mean number of clusters (per membrane) in the two-membrane model, as a function of the intra- and intermembrane protein interaction energies. Each of the two 100×100 square lattices hosted 500 proteins, a density of 0.05. (Bottom) a 3D plot; (Top) same data, viewed from above. Color coding helps to identify the range of parameters that give strong clustering: blue and purple colors correspond to fewer than 2 clusters per membrane in the ensemble. The model was run for 15,000 iterations. To reduce the statistical variation, $1/N_c$ was averaged over 5 runs.

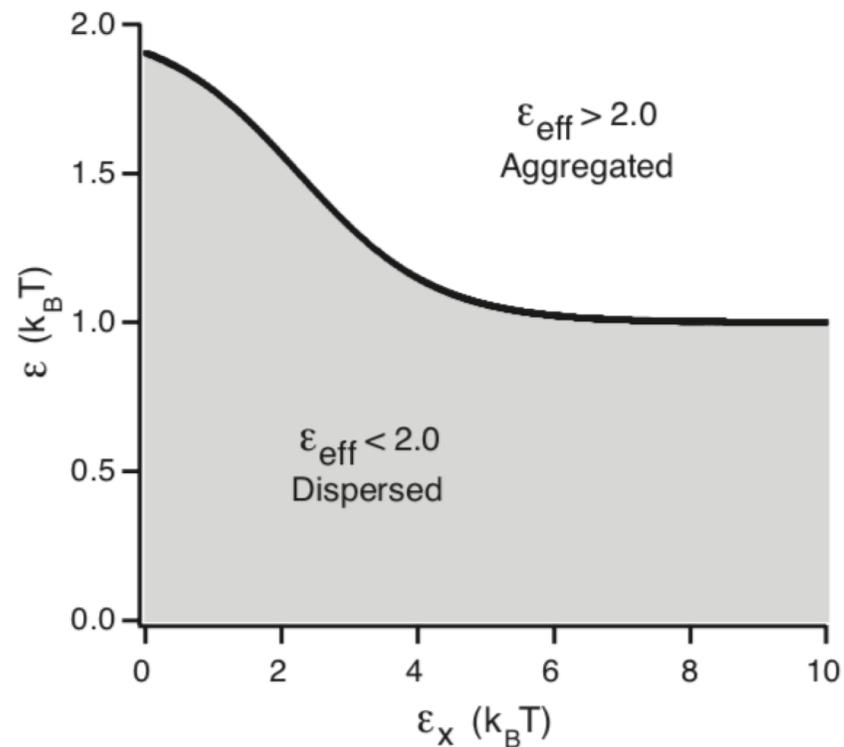


Fig. 6. The strengths of the intra- (ϵ) and intermembrane (ϵ_x) protein interaction energies (in $k_B T$) required to give an effective interprotein interaction of $2 k_B T$, according to a simple “mean field” estimate. $2 k_B T$ is the threshold for receptor aggregation at this concentration (5%). The line should be compared with the boundary between aggregated (blue) and dispersed (red) protein phases in Fig. 5, Top.

RuleBuilder Layout

The screenshot displays the RuleBuilder 1.40 Beta software interface. At the top is a menu bar with 'File', 'Edit', 'View', and 'Help'. Below it is a toolbar with various icons for drawing and editing. The main workspace is divided into several panels:

- Drawing Board:** The central workspace where a reaction rule is being constructed. It shows two reactants, labeled 'A' and 'B', each represented by a circle containing a smaller circle with the letter 'b'. These are followed by a plus sign and an arrow pointing to the products, which are two circles labeled 'A' and 'B' connected by a horizontal line. A yellow callout box explains: "The Drawing Board is where containers, components, edges and operators can be placed in order to create the molecules, species, reaction rules, observables, and patterns that form a BioNetGen model."
- Molecule Templates Palette:** A panel on the right side containing two molecule templates, labeled 'A' and 'B', each consisting of a circle with a smaller circle inside labeled 'b'. A yellow callout box states: "Defined objects, such as Molecule Templates, Species, and Reaction Rules are displayed in separate windows."
- Reaction Rules:** A panel at the bottom of the interface, currently empty.

At the bottom left, the text 'Object Manipulation Mode' is visible. The bottom of the window features a blue bar with a small window icon.

Adding Containers and Components

The screenshot displays the Rule Builder 1.40 Beta software interface. The main window is titled "Rule Builder 1.40 Beta - gsg_example" and features a menu bar with "File", "Edit", "View", and "Help". Below the menu bar is a toolbar containing various icons for drawing and editing. The central area is the "Drawing Board", which currently shows a small circle containing the letter "c". A yellow callout box points to the "Add Container Mode" icon in the toolbar, with the text "Add Container Mode." and "A component is added by entering 'Add Components' mode on the toolbar and left-clicking in the Drawing Board." To the right of the Drawing Board are two vertical panels: "Molecule Templates Palette" and "Seed Species". At the bottom of the interface is a "Reaction Rules" panel, which is currently empty. The status bar at the very bottom indicates "Object Manipulation Mode".

Rule Builder 1.40 Beta - gsg_example

File Edit View Help

Drawing Board

Molecule Templates Palette

Seed Species

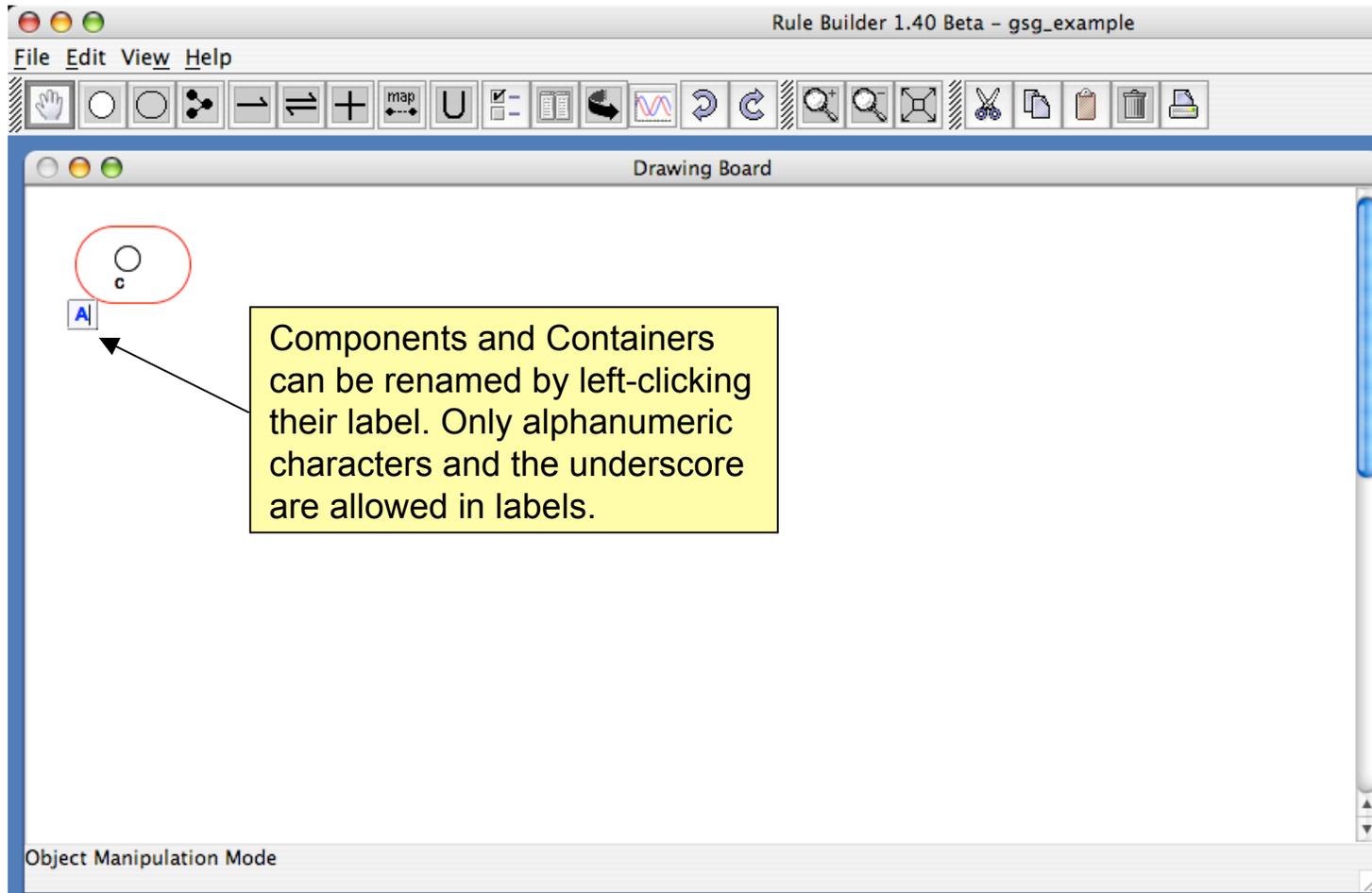
Reaction Rules

Object Manipulation Mode

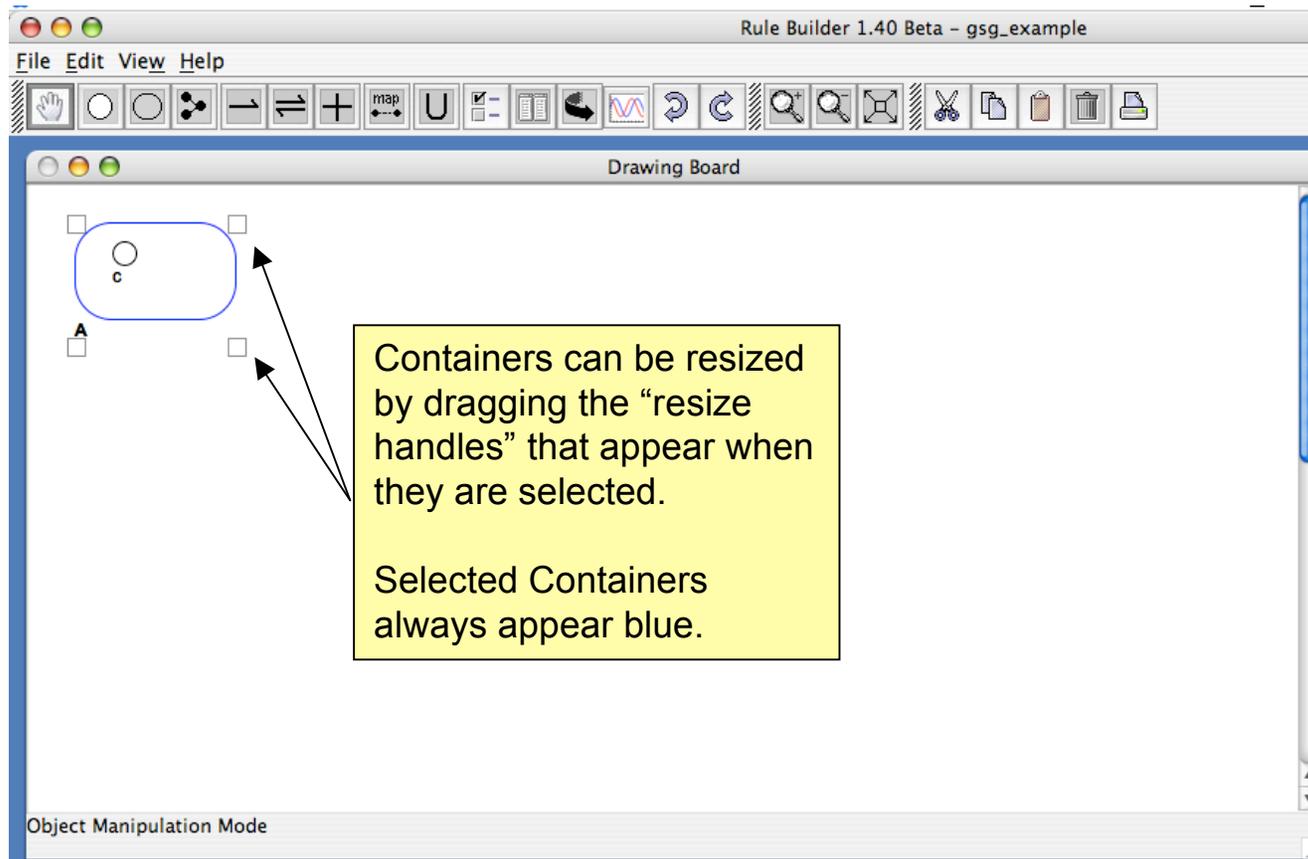
Add Container Mode.

A component is added by entering "Add Components" mode on the toolbar and left-clicking in the Drawing Board.

Renaming Components and Containers

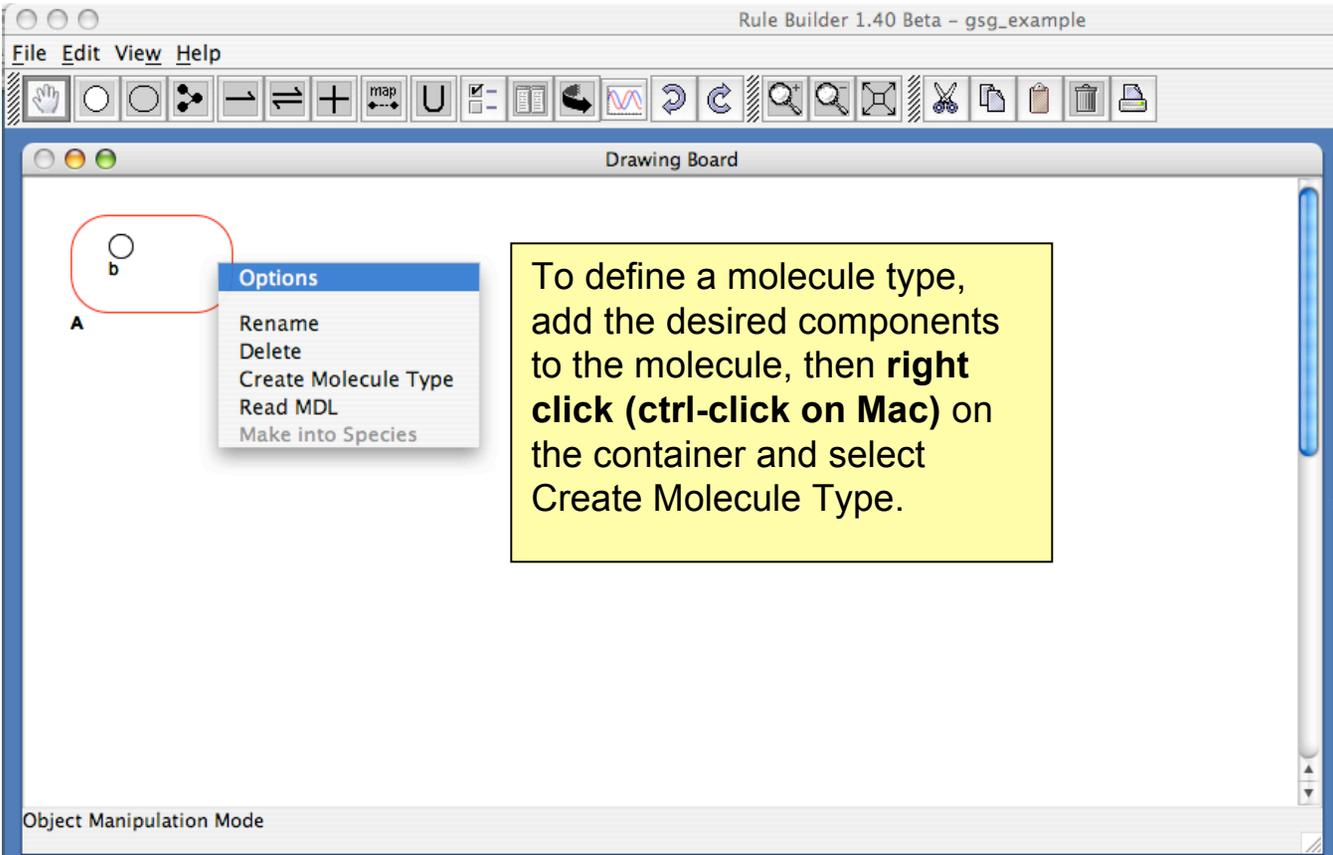


Resizing Containers



Creating Molecule Types

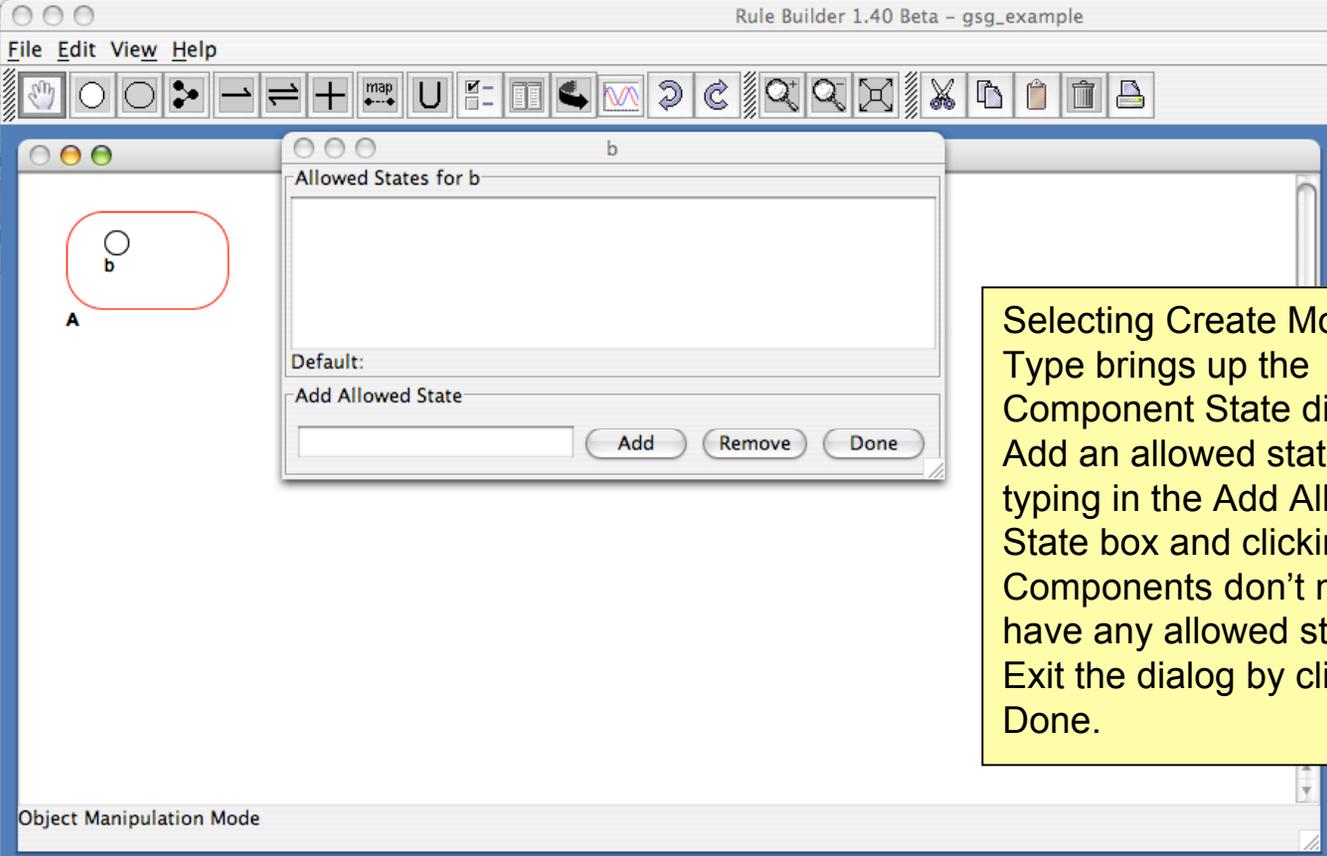
Molecules used in a model have to be defined and registered as a “Molecule Type” before they can be used in reaction rules and species.



The screenshot shows the Rule Builder 1.40 Beta software interface. The main window is titled "Drawing Board" and contains a drawing area. In the drawing area, a small circle labeled "b" is highlighted with a red oval. A context menu is open over this circle, listing the following options: "Options", "Rename", "Delete", "Create Molecule Type", "Read MDL", and "Make into Species". The "Create Molecule Type" option is highlighted in blue. A yellow text box on the right side of the drawing board contains the following text: "To define a molecule type, add the desired components to the molecule, then **right click (ctrl-click on Mac)** on the container and select **Create Molecule Type**." The software's menu bar includes "File", "Edit", "View", and "Help". The status bar at the bottom left indicates "Object Manipulation Mode".

Setting Allowed Component States

Components may take on different states to indicate conformation or covalent modification, such as phosphorylation.



The screenshot shows the 'Rule Builder 1.40 Beta - gsg_example' application window. The main workspace contains a diagram with a component labeled 'b' inside a rounded rectangle labeled 'A'. A dialog box titled 'Allowed States for b' is open, showing a list of allowed states (currently empty), a 'Default:' field, and an 'Add Allowed State' section with a text input box and 'Add', 'Remove', and 'Done' buttons. The status bar at the bottom indicates 'Object Manipulation Mode'.

Selecting Create Molecule Type brings up the Component State dialog. Add an allowed state by typing in the Add Allowed State box and clicking Add. Components don't need to have any allowed states. Exit the dialog by clicking Done.

Identifying Valid and Invalid Molecules

The screenshot displays the Rule Builder 1.40 Beta software interface. The main window is titled "Rule Builder 1.40 Beta - gsg_example" and contains a "Drawing Board" and a "Molecule Templates Palette".

The Drawing Board shows three molecules labeled A, B, and A. Molecule A (top left) is a rounded rectangle with a small circle containing the letter 'b' inside, outlined in green. Molecule B (top right) is a rounded rectangle with a small circle containing the letter 'b' inside, outlined in red. Molecule A (bottom left) is a rounded rectangle with a dashed green outline.

Yellow callout boxes provide the following information:

- Top right: "Molecule Types appear here" (pointing to the Molecule Templates Palette).
- Left of molecule A (top left): "Containers matching valid types are **green**."
- Right of molecule B (top right): "Containers not matching a valid type are **red**."
- Right of molecule A (bottom left): "Dashed line indicates an incomplete match."

The Molecule Templates Palette on the right shows a single molecule template labeled A, which is a rounded rectangle with a small circle containing the letter 'b' inside. Below it is a "Seed Species" section.

The bottom status bar indicates "Object Manipulation Mode".

Copying Objects with the Selection Box

RuleBuilder 1.45 Beta

File Edit View Help

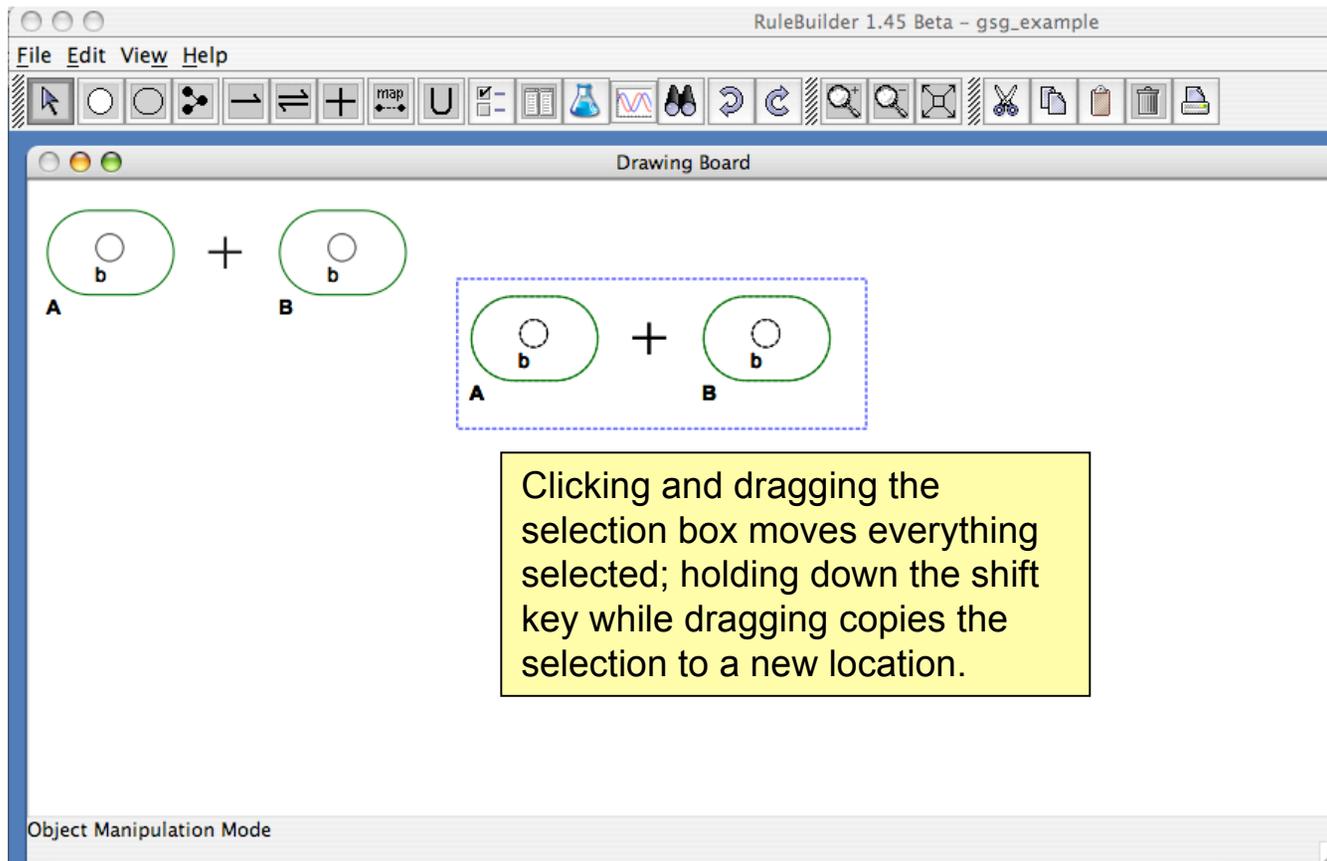
Drawing Board

A + B

Object Manipulation Mode

Draw a box around objects on the Drawing Board to select them. Partially enclosed objects are not selected.

Copying Objects with the Selection Box



Creating a Reaction Rule

Reaction rules are created by arranging containers and operators to construct a formula for the reaction.

The screenshot displays the 'Rule Builder 1.40 Beta - gsg_example' application window. The main workspace is titled 'Drawing Board' and contains two containers, 'A' and 'B', each containing a small circle with a letter inside. Container 'A' contains a circle with the letter 'b', and container 'B' contains a circle with the letter 'a'. A red '+' operator is placed between containers 'A' and 'B', and a red '-' operator is placed to the right of container 'B'. A yellow callout box points to the '+' operator with the text: 'The '+' operator separates reactants or products in a list.' Another yellow callout box points to the '-' operator with the text: 'The arrow operator separates reactants and products.' The interface includes a menu bar (File, Edit, View, Help), a toolbar with various icons, and two side panels: 'Molecule Templates Palette' and 'Seed Species'. The status bar at the bottom indicates 'Object Manipulation Mode'.

Creating a Reaction Rule

The type of arrow determines whether a reaction is reversible or irreversible.

The screenshot displays the Rule Builder 1.40 Beta software interface. The main window is titled "Rule Builder 1.40 Beta - gsg_example" and features a menu bar with "File", "Edit", "View", and "Help". Below the menu bar is a toolbar with various icons, including a red box highlighting the irreversible arrow icon (a single arrow pointing right) and the reversible arrow icon (two arrows pointing in opposite directions). The central "Drawing Board" shows a chemical reaction rule: a rounded rectangle labeled "A" containing a small circle with the letter "b" inside, followed by a plus sign, another rounded rectangle labeled "B" containing a small circle with the letter "a" inside, and a single arrow pointing to the right. To the right of the drawing board are two panels: "Molecule Templates Palette" and "Seed Species". The "Molecule Templates Palette" contains two rounded rectangles, one labeled "A" with a circle containing "b" and another labeled "B" with a circle containing "a". The "Seed Species" panel is currently empty. At the bottom left of the interface, the text "Object Manipulation Mode" is visible.

Defining Products

RuleBuilder 1.45 Beta - gsg_example

File Edit View Help

Drawing Board

A + B → A B

Use Add Edges to create a bond between the components

Object Manipulation Mode

Defining Products

RuleBuilder 1.45 Beta - gsg_example

File Edit View Help

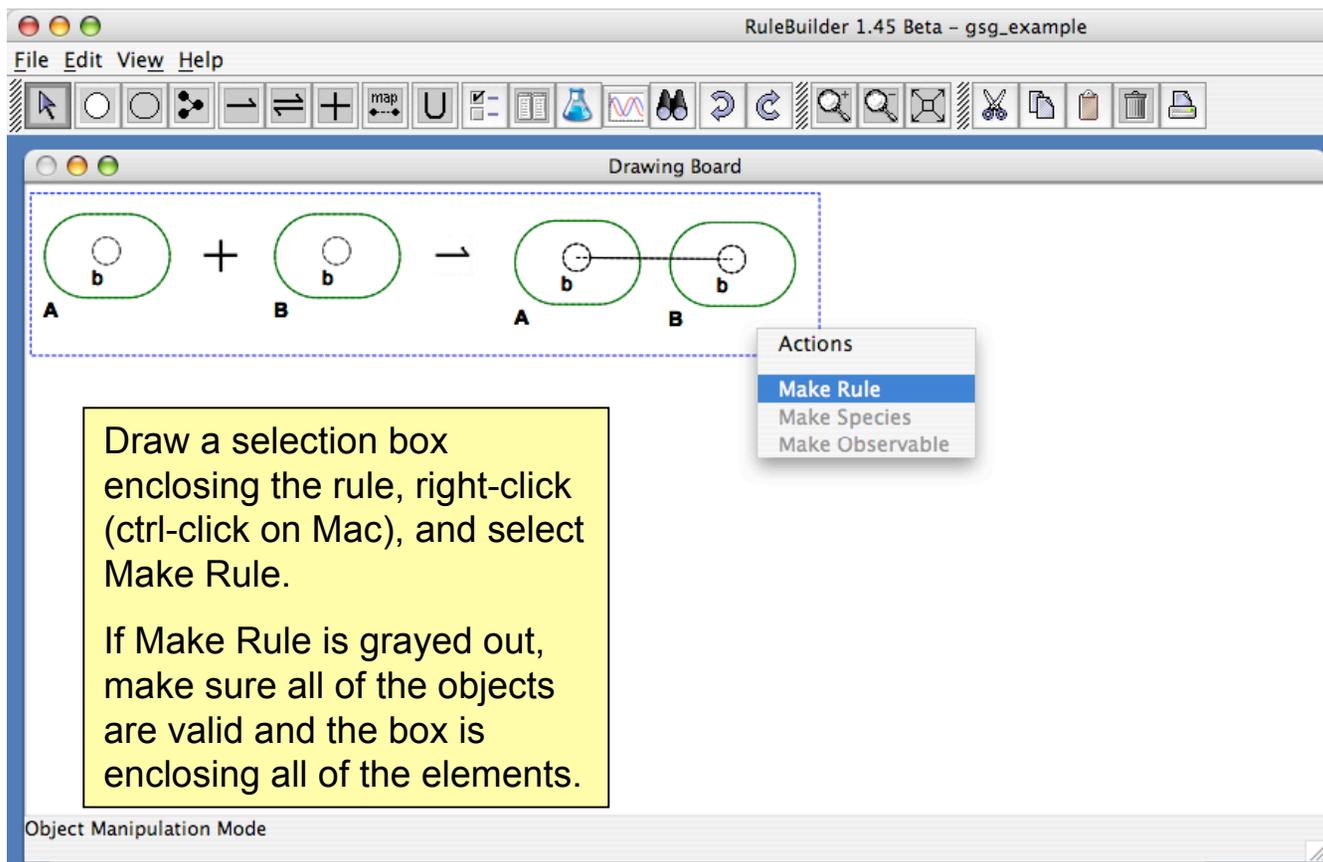
Drawing Board

A + B → A B

Create the bond by clicking on the two components to be linked.

Add Edge Mode

Creating the Rule



The screenshot shows the RuleBuilder 1.45 Beta software interface. The main window is titled "RuleBuilder 1.45 Beta - gsg_example" and contains a "Drawing Board" window. The drawing board displays a chemical rule diagram: two green ovals labeled "A" and "B" (each containing a smaller white circle with a "b" inside) are on the left, separated by a plus sign and an arrow. On the right, the same two ovals are shown connected by a horizontal line, representing the product of the rule. A blue dashed selection box encloses the entire diagram. A context menu is open over the diagram, listing "Actions": "Make Rule" (highlighted in blue), "Make Species", and "Make Observable".

File Edit View Help

RuleBuilder 1.45 Beta - gsg_example

Drawing Board

A + B → A B

Actions

- Make Rule
- Make Species
- Make Observable

Draw a selection box enclosing the rule, right-click (ctrl-click on Mac), and select Make Rule.

If Make Rule is grayed out, make sure all of the objects are valid and the box is enclosing all of the elements.

Object Manipulation Mode

Make Rule Dialog

The screenshot shows the 'RuleBuilder 1.45 Beta - gsg_example' application window. The 'Make Rule Dialog' is open, displaying the following fields:

- Rule Name:
- Label: Rule1
- Forward Rate Name: kp1
- Rate:
- BNGL Annotation:

The 'Drawing Board' shows a reaction diagram with a reactant 'b' (a circle with a smaller circle inside) and a product 'b' (a circle with a smaller circle inside). The reaction is labeled 'A' and '+'. The 'Rate' field in the dialog is highlighted with a yellow box and an arrow pointing to it.

Set Rule Name, rate constants, and optional annotation in the dialog box.

For a parameter being used for the first time, set a numerical value in the Rate box.

Object Manipulation Mode

Reaction Rules Window

The screenshot displays the RuleBuilder 1.45 Beta software interface. The main window is titled "RuleBuilder 1.45 Beta - gsg_example" and contains a "Drawing Board" and a "Reaction Rules" window. The "Drawing Board" shows a chemical reaction rule: two separate molecules, each consisting of a circle labeled 'b' inside an oval labeled 'A' or 'B', react to form a single molecule where two 'b' circles are connected by a line, each still within its respective 'A' or 'B' oval. The "Reaction Rules" window at the bottom shows the same reaction rule, but with the rate constant "kp1" above the reaction arrow and the label "Rule1" below the reactants. A yellow callout box points to the "Reaction Rules" window with the text "Rule now appears in the Reaction Rules Window." The interface also includes a menu bar (File, Edit, View, Help), a toolbar with various icons, and two side panels: "Molecule Templates Palette" and "Seed Species".

Rule now appears in the Reaction Rules Window.

Defining Seed Species

The network is defined by applying the reaction rules to a set of seed species.

RuleBuilder 1.45 Beta - gsg_example

File Edit View Help

Drawing Board

Molecule Templates Palette

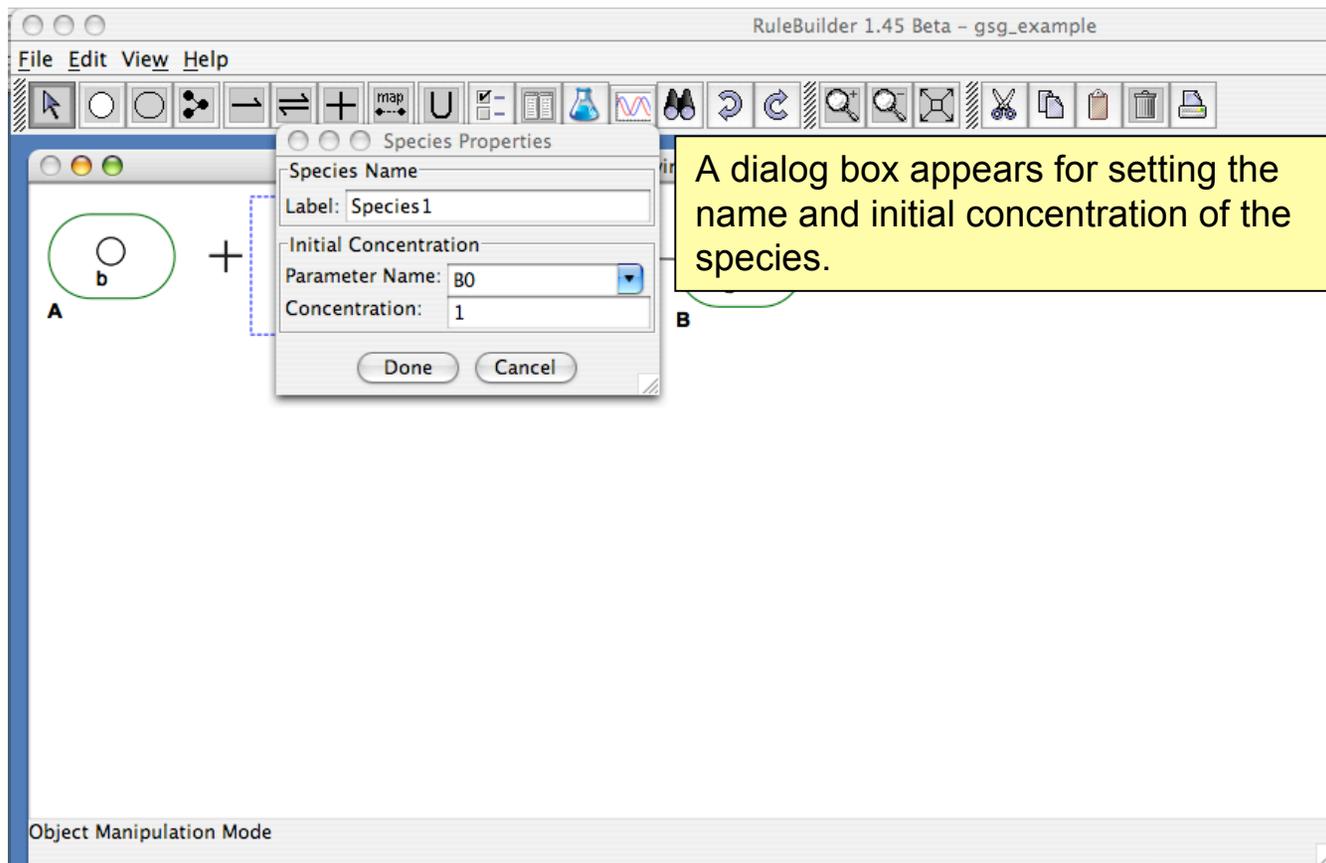
Seed Species

Object Manipulation M

Draw a selection box around a connected set of molecules and right-click (ctrl-click on Mac) to define a species.

All items in selection box should have solid green lines, indicating the the molecules are fully defined.

Species Dialog Box



Seed Species Window

The screenshot displays the RuleBuilder 1.45 Beta interface. The main window, titled "RuleBuilder 1.45 Beta - gsg_example", contains a "Drawing Board" and a "Molecule Templates Palette".

The Drawing Board shows a chemical reaction: two separate molecules, labeled A and B, each consisting of a small circle 'b' inside a larger oval, are added together (+). An arrow points to the product, which consists of two such molecules, A and B, connected by a horizontal line.

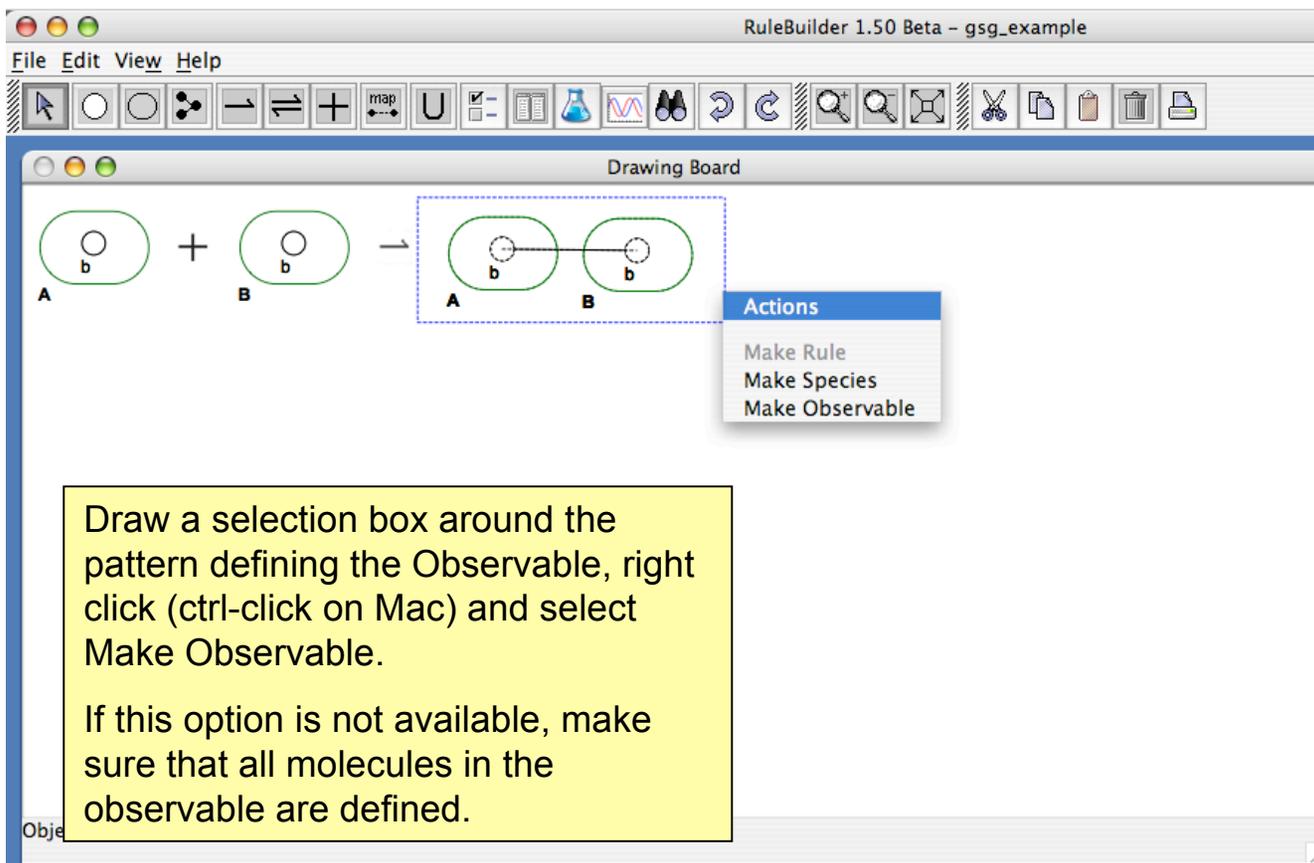
The Molecule Templates Palette on the right shows two templates, A and B, identical to the reactants.

Below the palette is the "Seed Species" window, which contains two entries: "A Species0" and "B Species1". Each entry shows a molecule template. A yellow callout box with the text "New species appears in the Seed Species Window." has an arrow pointing to the "B Species1" entry.

At the bottom left of the interface, the text "Object Manipulation Mode" is visible.

Defining Observables

Observables are concentration sums over species with particular properties and correspond to model outputs, such as total phosphorylation of a protein.



The screenshot shows the RuleBuilder 1.50 Beta software interface. The main window is titled "RuleBuilder 1.50 Beta - gsg_example" and contains a "Drawing Board" window. The drawing board displays a chemical reaction: two molecules labeled 'A' and 'B' (each represented by a green oval with a smaller white circle inside labeled 'b') react to form a complex of two such molecules connected by a horizontal line. A dashed blue selection box is drawn around the product complex. A context menu is open over the selection box, titled "Actions", with three options: "Make Rule", "Make Species", and "Make Observable".

Draw a selection box around the pattern defining the Observable, right click (ctrl-click on Mac) and select Make Observable.

If this option is not available, make sure that all molecules in the observable are defined.

Make Observables Dialog

RuleBuilder 1.50 Beta - gsg_example

File Edit View Help

Properties

Name:

Type: Molecules

Done

Set Rule Name and Type.

Type **Molecules** weights the concentration of each matching species by the number of times the defined pattern matches the species.

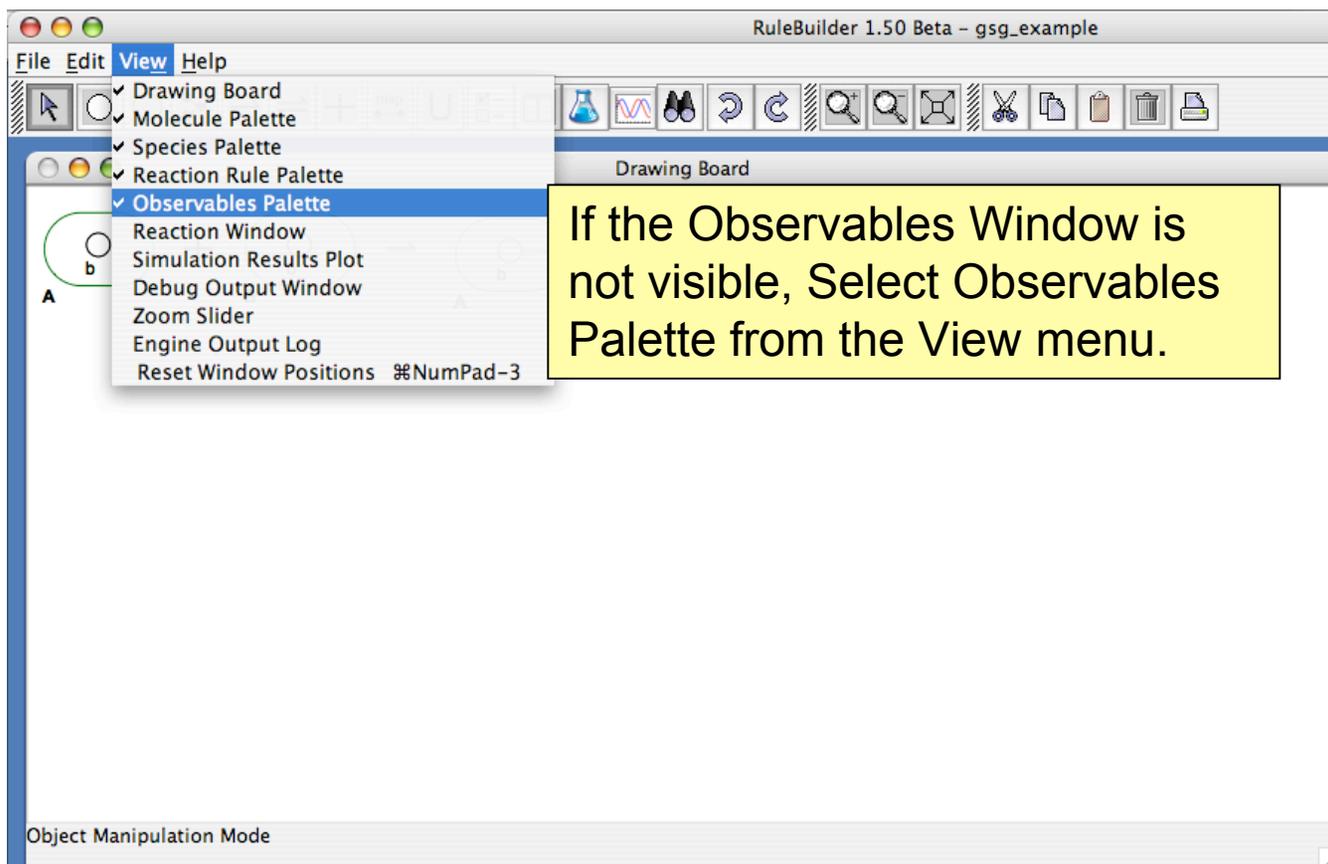
Use this for quantities like total phosphorylation of a site on a protein or total number of receptors in aggregates.

Type **Species** gives unit weight to the concentration of each matching species.

Use this type to get the concentration of complexes of a particular type.

Object Manipulation Mode

Observables Window



Observables Window

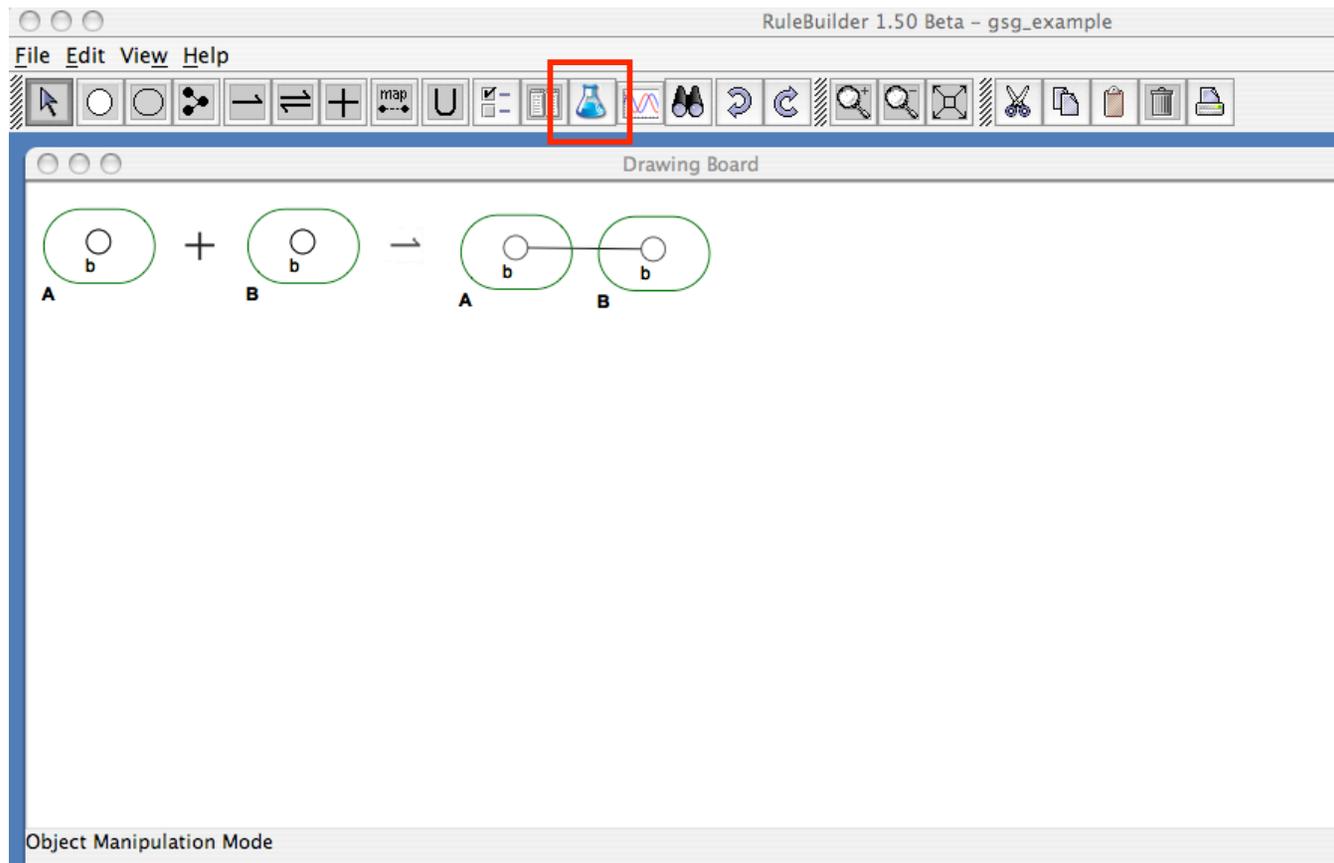
The screenshot displays the RuleBuilder 1.50 Beta software interface. The main window is titled "RuleBuilder 1.50 Beta - gsg_example" and contains several panels:

- File Edit View Help**: A menu bar at the top.
- Drawing Board**: A central workspace showing a chemical reaction rule. On the left, two separate molecules labeled "A" and "B" are shown, each consisting of a circle with a smaller circle inside and the letter "b" below it. An arrow points to the right, where the two molecules are now connected by a horizontal line, forming a dimer labeled "A" and "B".
- Molecule Templates Palette**: A panel on the right side containing two templates, "A" and "B", identical to the molecules in the Drawing Board.
- Seed Species**: A panel below the Molecule Templates Palette containing two species, "Species0" and "Species1", also identical to the molecules in the Drawing Board.
- Object Manipulation Mode**: A horizontal bar below the Drawing Board.
- Reaction Rules**: A panel at the bottom left showing the reaction rule "Rule1" with the same diagram as the Drawing Board.
- Observables**: A panel at the bottom right, highlighted with a yellow background, containing the dimer molecule from the Drawing Board. It is labeled "A" and "B" and "AB".

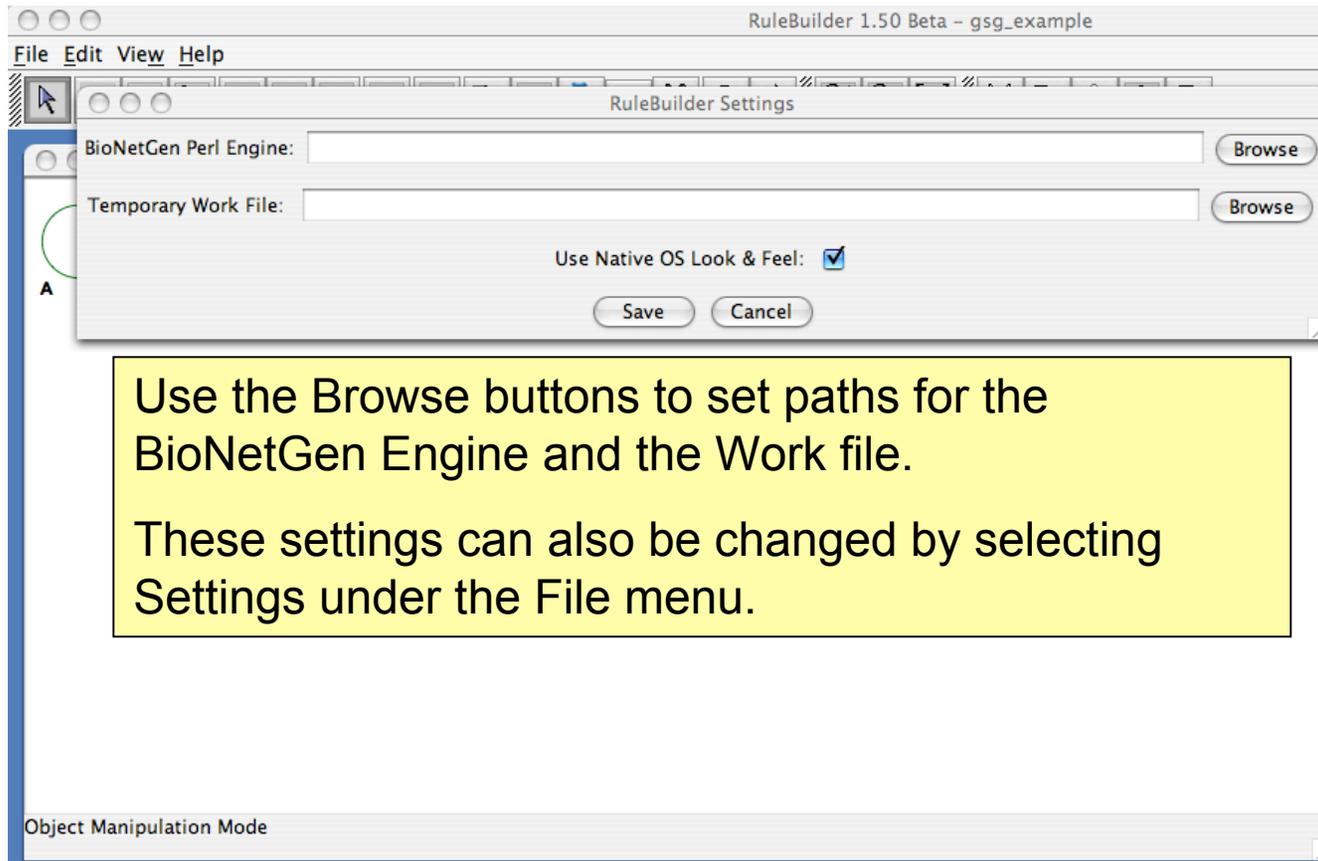
A yellow box with the text "The Observables Window" is overlaid on the Reaction Rules panel.

Running the Model

Once Reaction Rules, Seed Species, and Observables (optional) have been defined, the model can be simulated by pressing Run BioNetGen button.

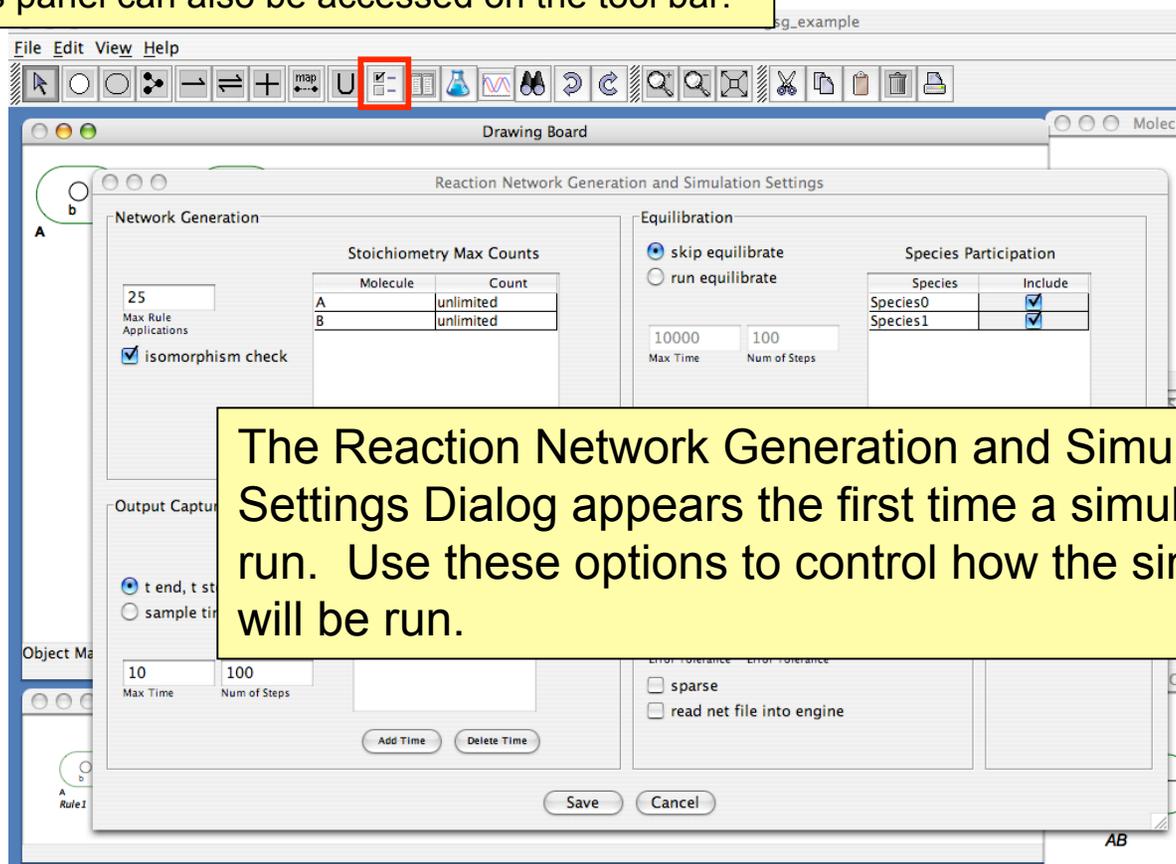


BioNetGen Engine Settings



The SimConfig Panel

This panel can also be accessed on the tool bar.



The Reaction Network Generation and Simulation Settings Dialog appears the first time a simulation is run. Use these options to control how the simulation will be run.

The SimConfig Panel

RuleBuilder 1.50 Beta - gsg_example

File Edit View Help

Drawing Board

Reaction Network Generation and Simulation Settings

Network Generation

25
Max Rule Applications

isomorphism check

Molecule	Count
A	unlimited
B	unlimited

Equilibration

skip equilibrate
 run equilibrate

10000 100
Max Time Num of Steps

Species Participation

Species	Include
Species0	<input checked="" type="checkbox"/>
Species1	<input checked="" type="checkbox"/>

Options

SBML output

1e-12 1e-12
Abs Integration Error Tolerance Rel Integration Error Tolerance

sparse
 read net file into engine

10 100
Max Time Num of Steps

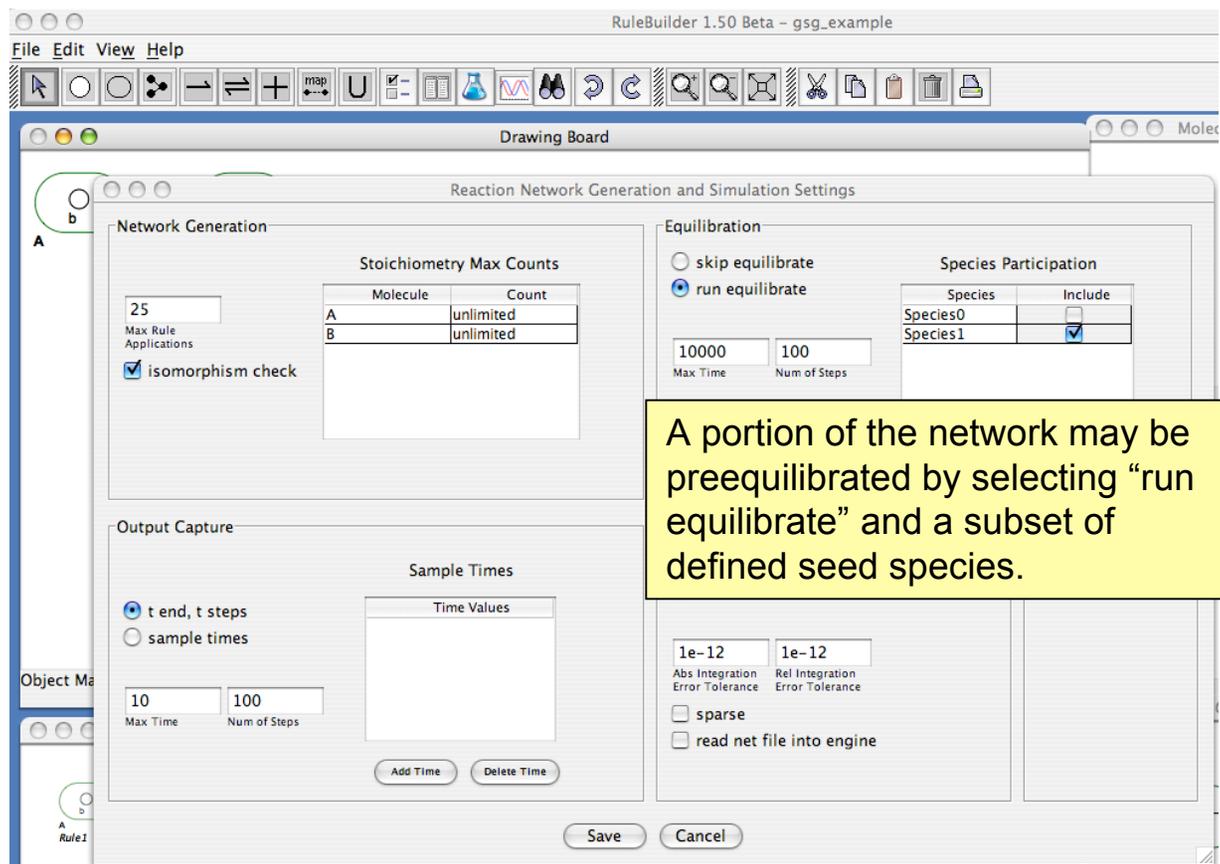
Add Time Delete Time

Save Cancel

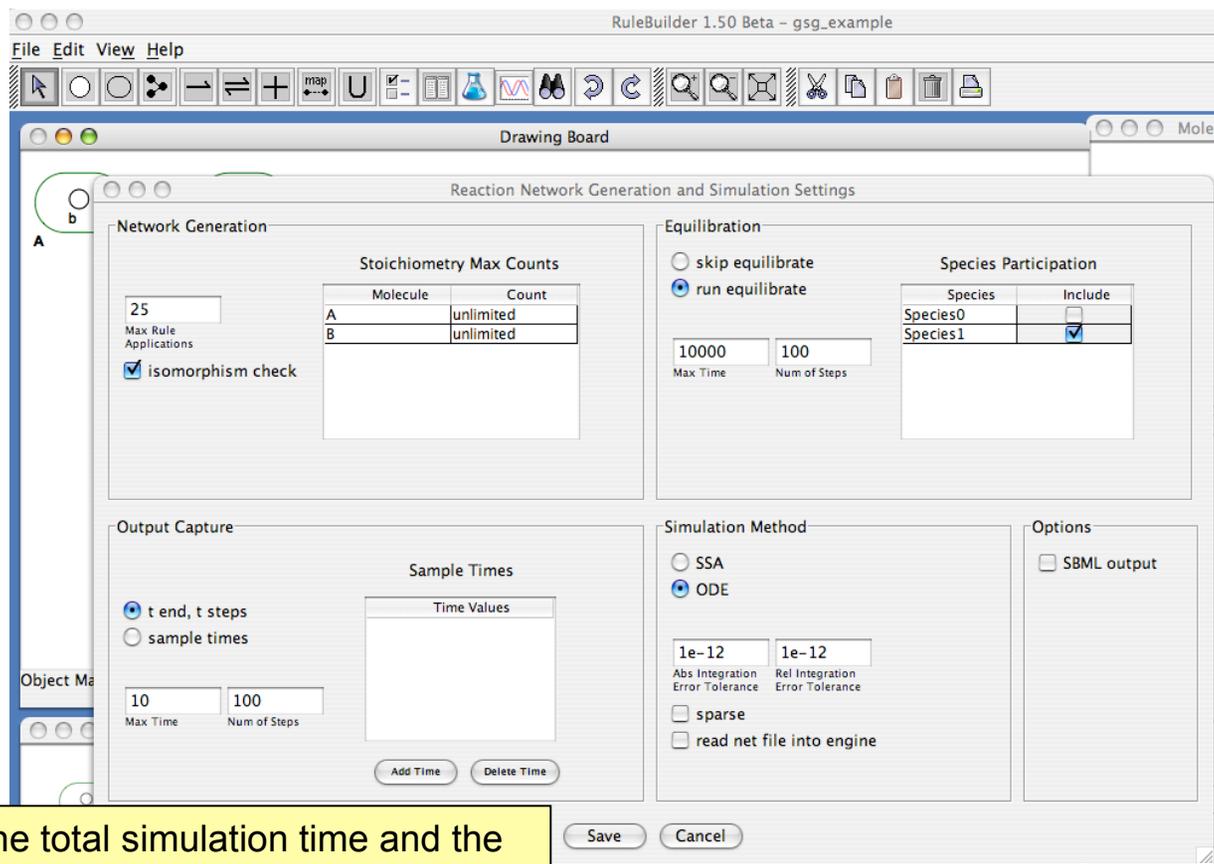
AB

The size of complexes and the reaction network can be limited by setting maximum values for the stoichiometry of molecules.

The SimConfig Panel

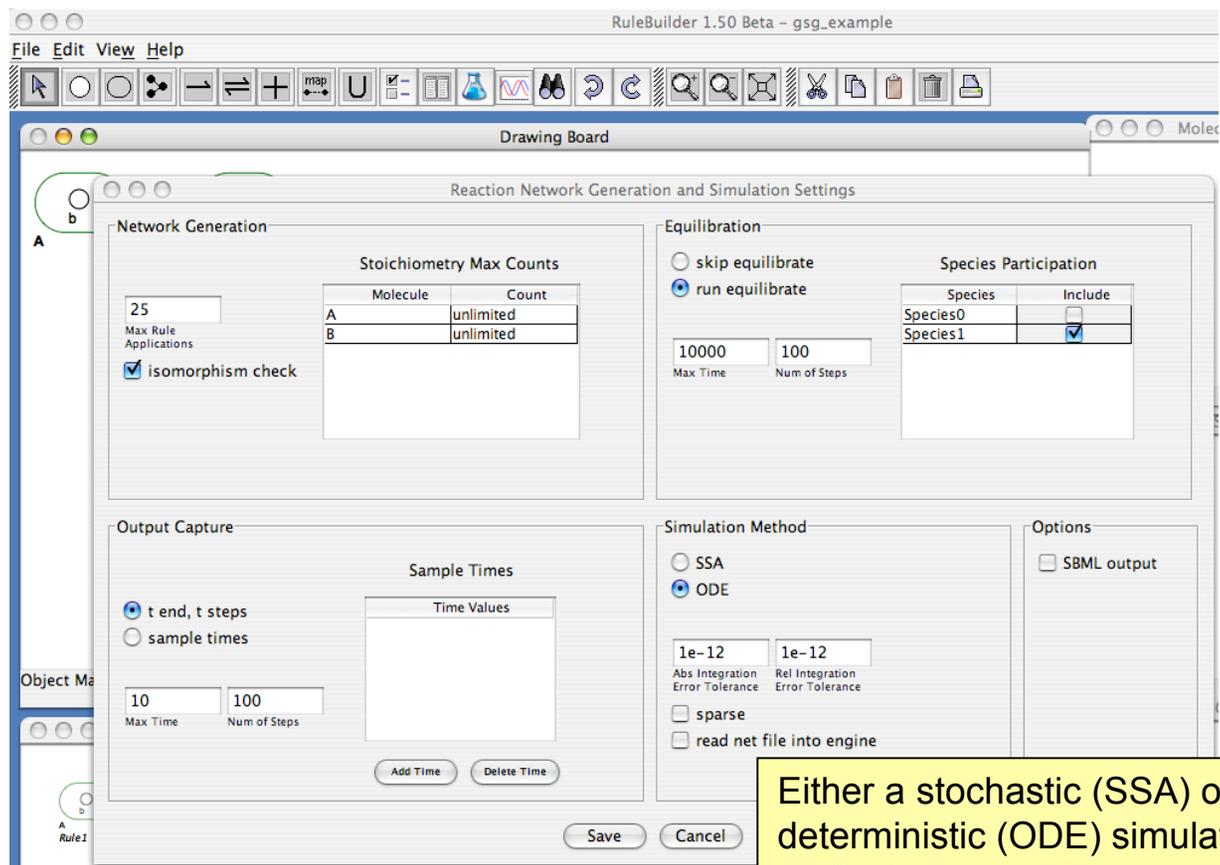


The SimConfig Panel

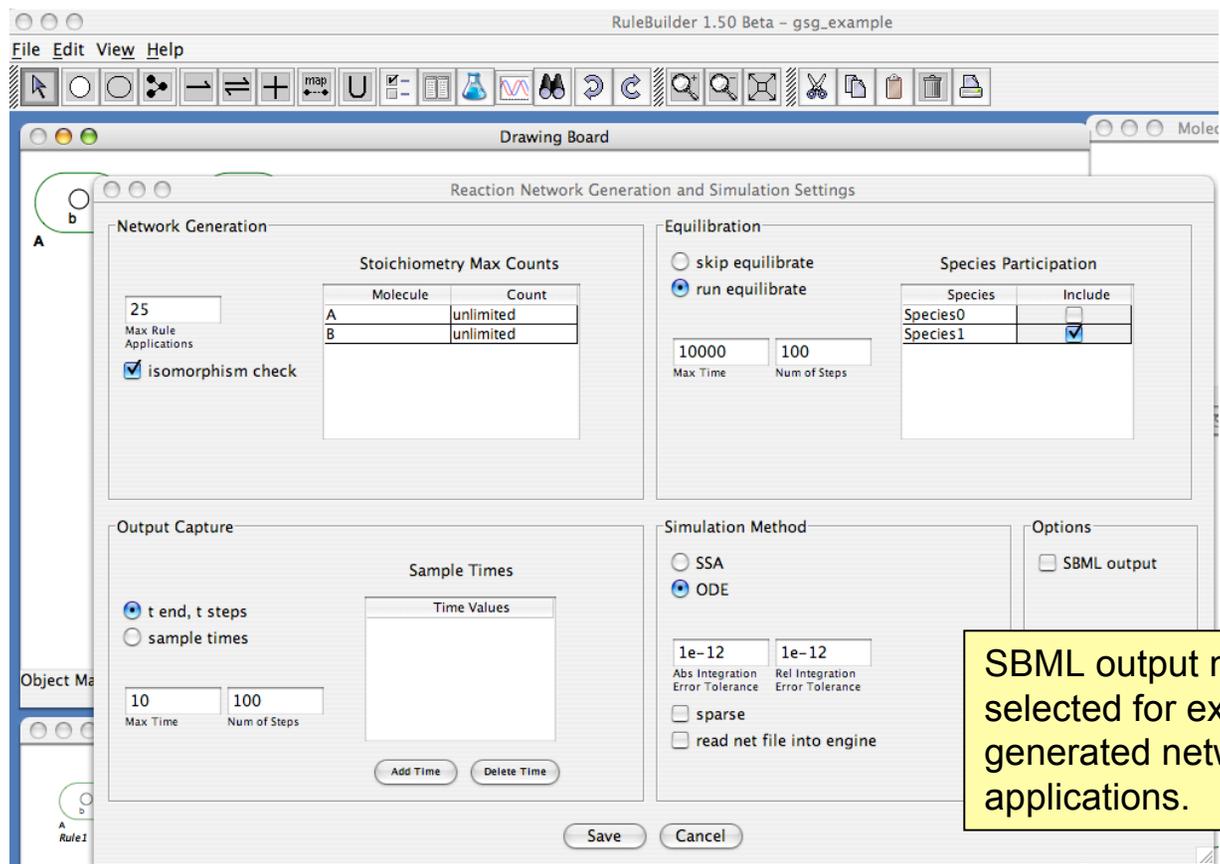


The total simulation time and the times at which concentrations are sampled are set here.

The SimConfig Panel



The SimConfig Panel



The Log Window

Running the simulation brings up the BioNetGen Output Log or Log Window

The screenshot shows a software window titled "RuleBuilder 1.40 Beta - gsg_example" with a toolbar. Below it is a "Drawing Board" window containing a "BioNetGen Output Log" window. The log window displays a table of data and summary statistics.

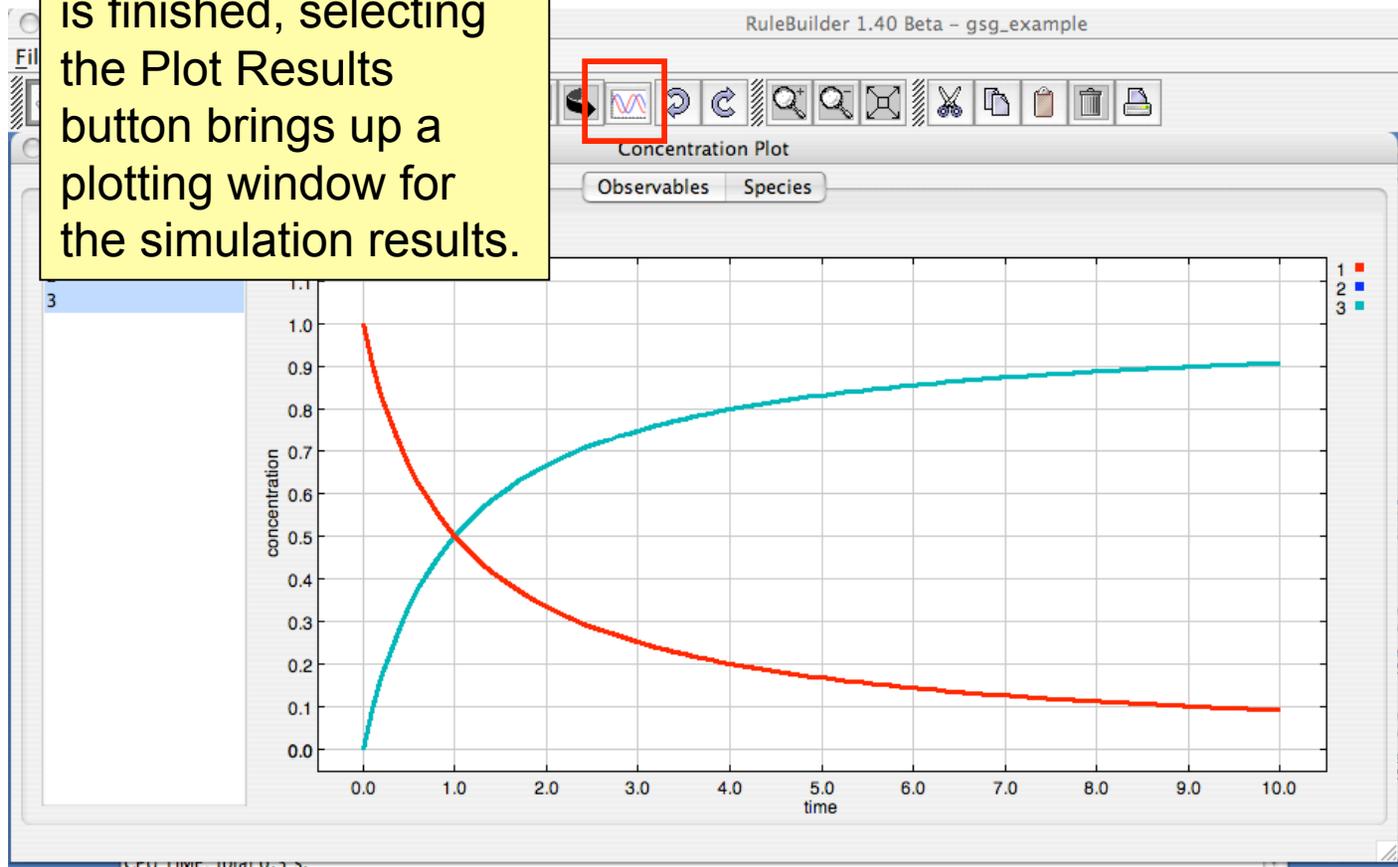
7.70	594	646
7.80	596	648
7.90	598	650
8.00	600	652
8.10	602	654
8.20	604	656
8.30	606	658
8.40	608	663
8.50	611	666
8.60	613	668
8.70	615	670
8.80	617	672
8.90	619	674
9.00	621	676
9.10	623	678
9.20	625	680
9.30	627	682
9.40	630	685
9.50	632	687
9.60	634	689
9.70	636	691
9.80	638	693
9.90	640	695
10.00	642	697

Time course of concentrations written to file /Users/faeder/shared/Projects/BioNetGen_develop/temp.cdat.
Propagation took 0.00 CPU seconds
Program times: 0.01 CPU s 0.00 clock s
Updating species concentrations from /Users/faeder/shared/Projects/BioNetGen_develop/temp.cdat
CPU TIME: simulate_ode 0.0 s.
Finished processing file /Users/faeder/shared/Projects/BioNetGen_develop/temp.bngl
CPU TIME: total 0.3 s.

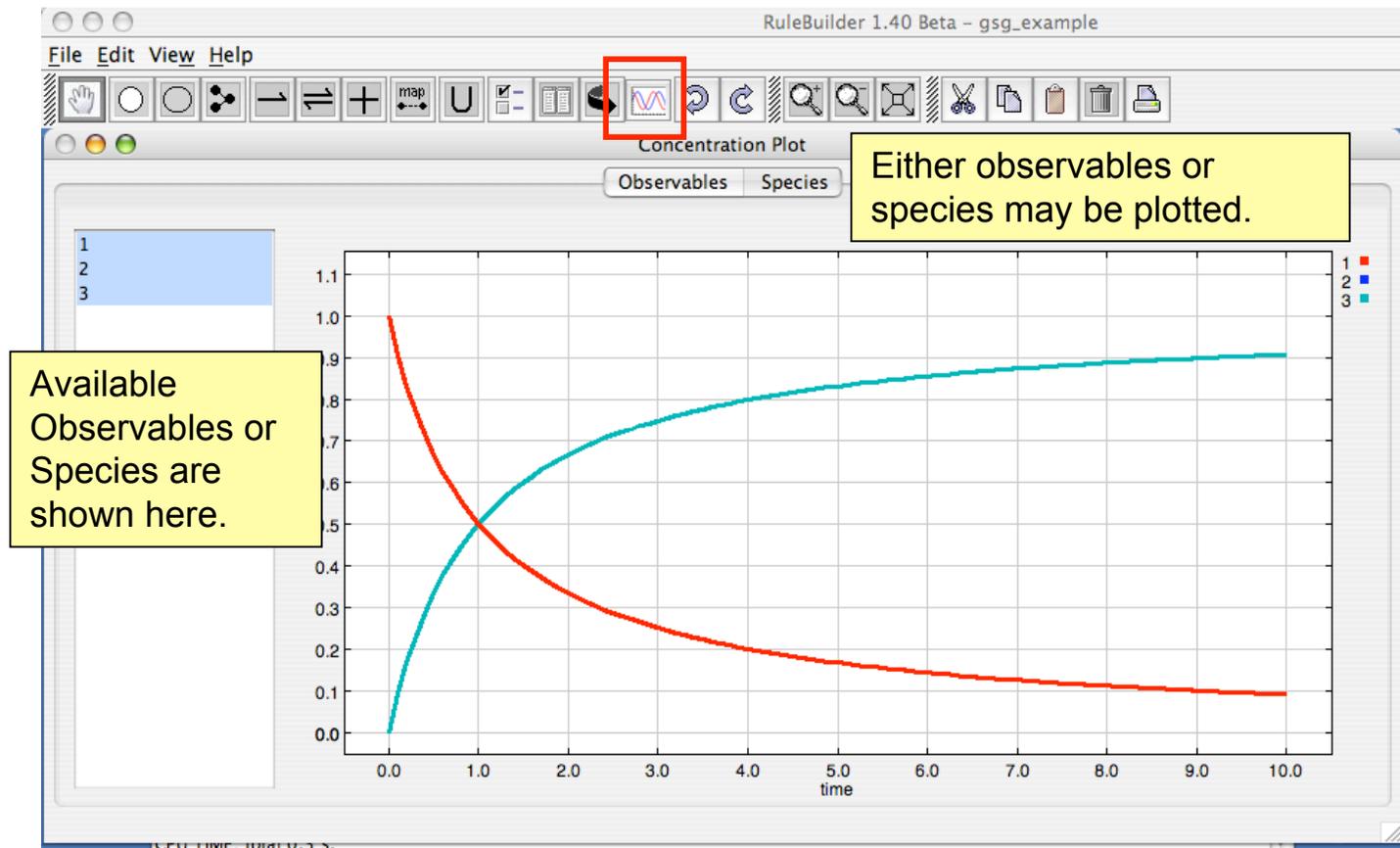
The Log Window displays the output of BioNetGen.

Plotting the Results

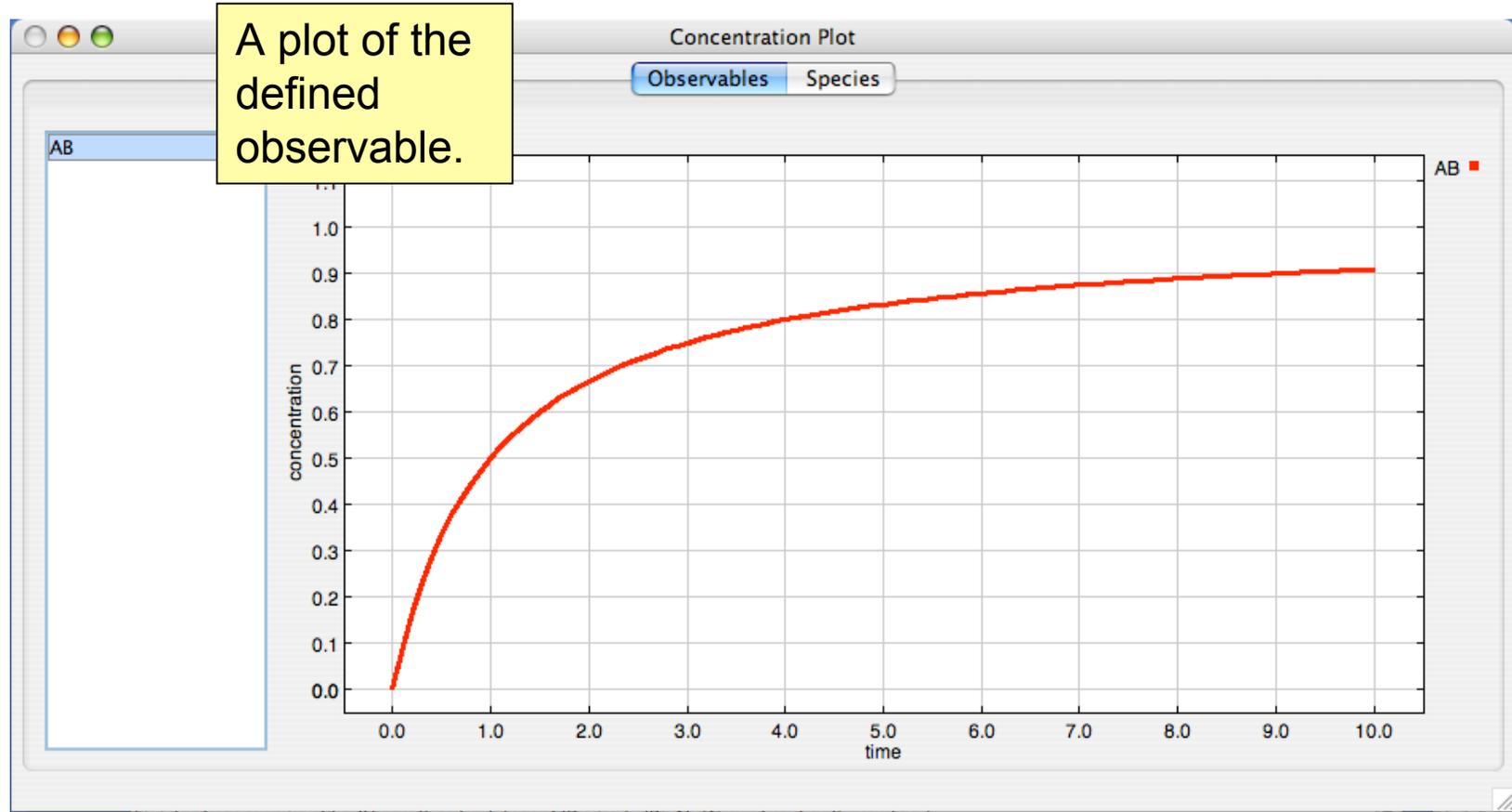
Once the simulation is finished, selecting the Plot Results button brings up a plotting window for the simulation results.

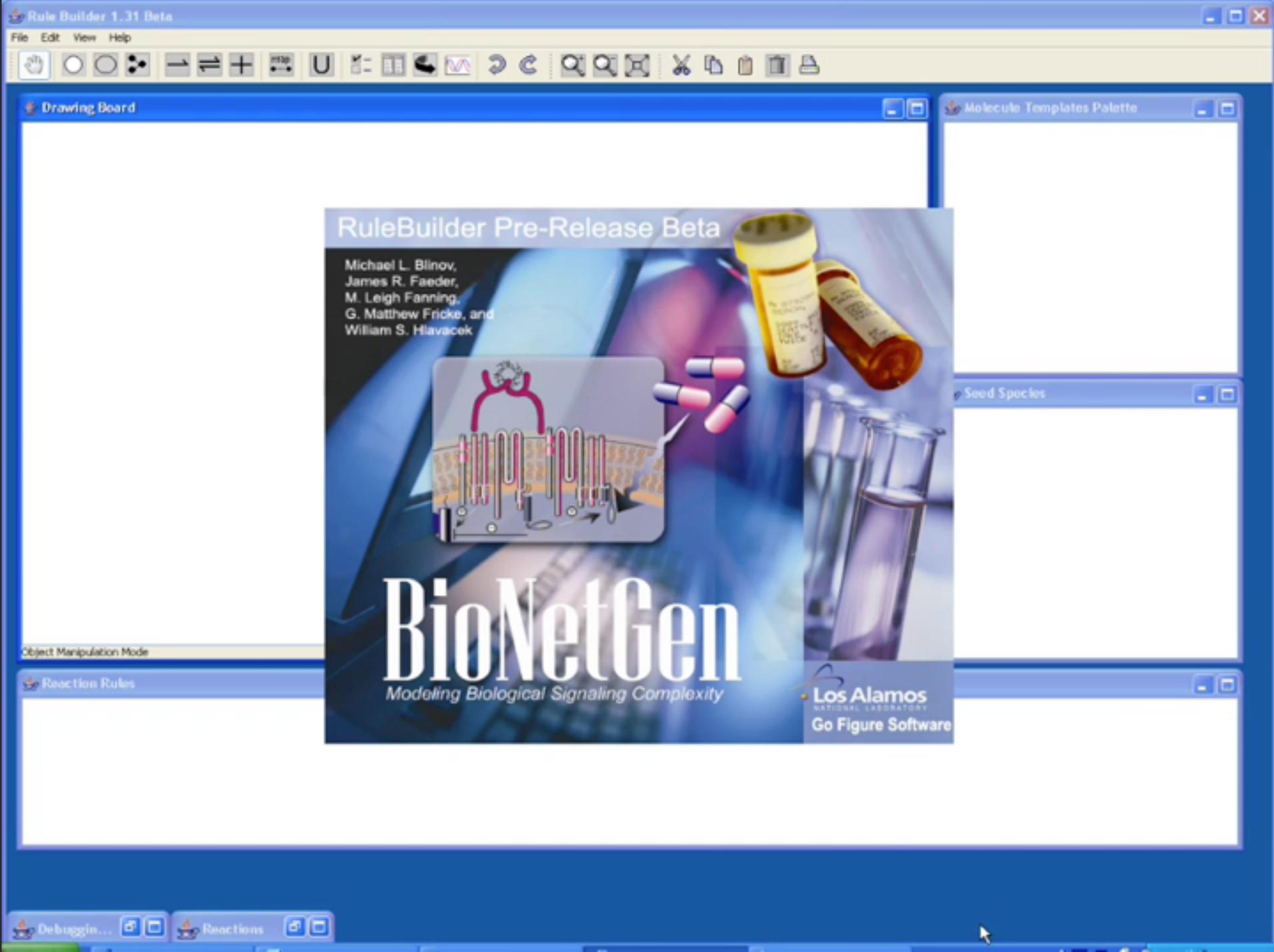


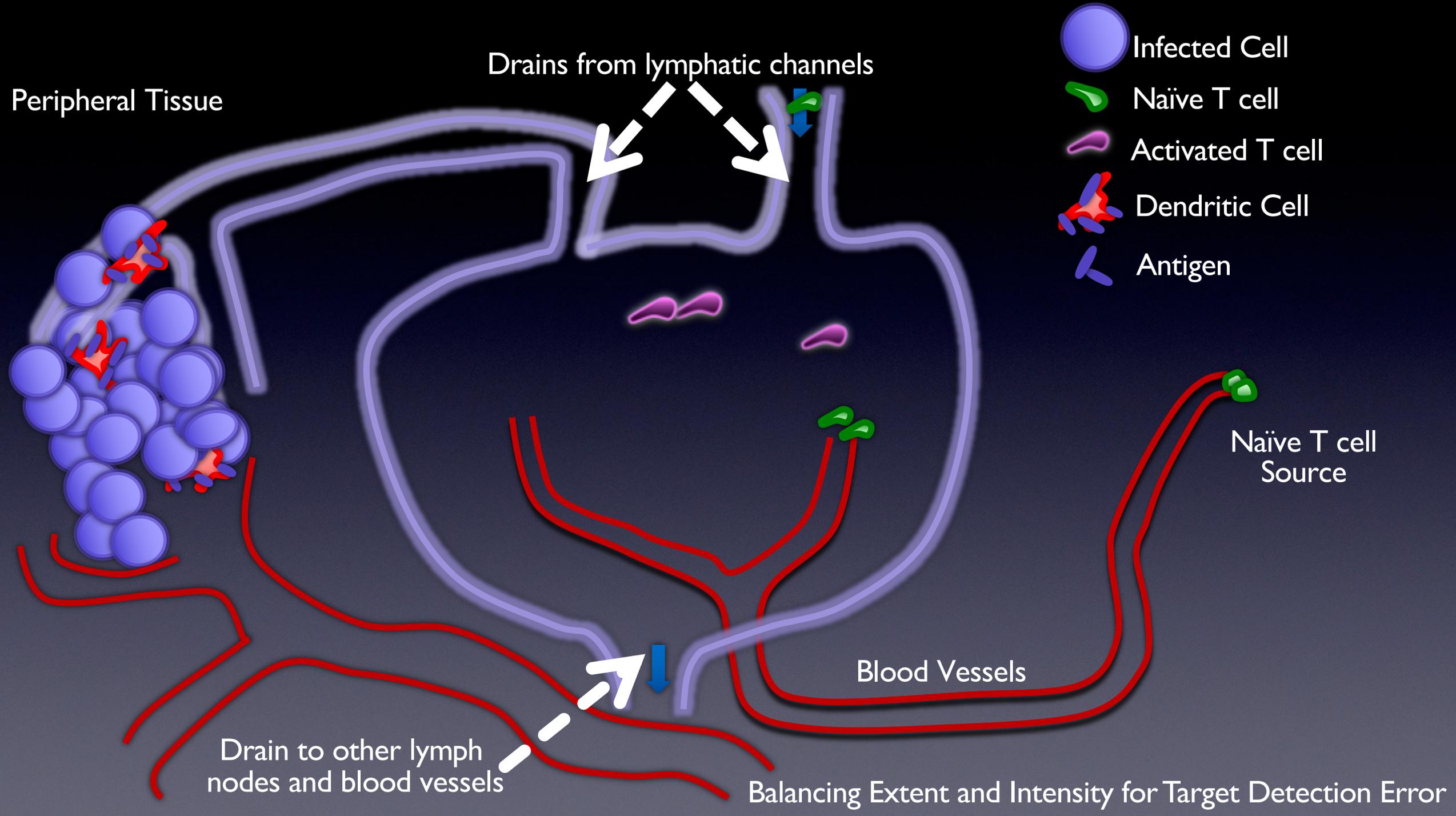
Plotting the Results

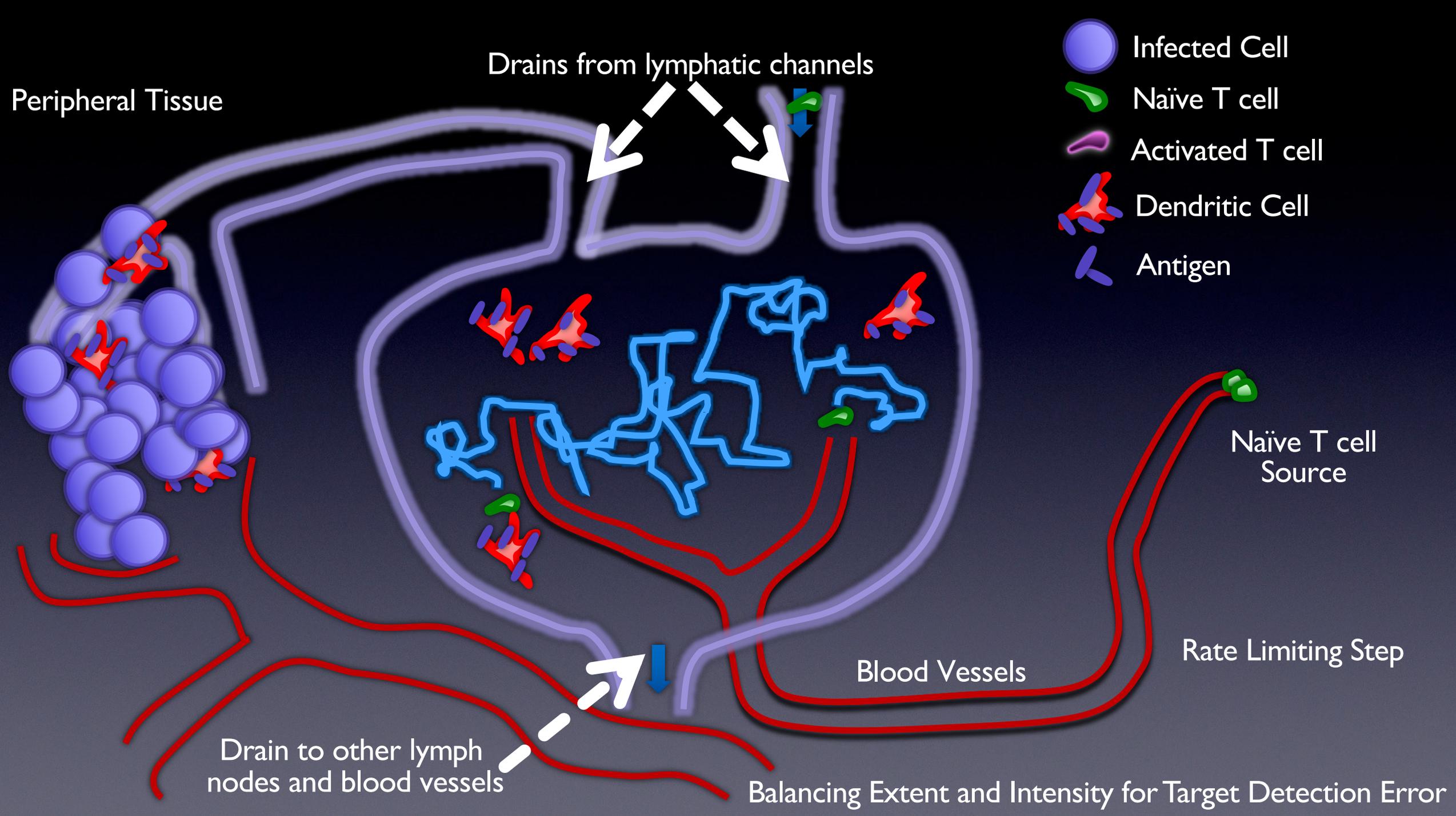


Plotting the Results









The Needle in a Haystack

Lymph nodes have a volume 10^6 times that of T cells.

100k T cells and 100k DCs. Small set of these are cognate.

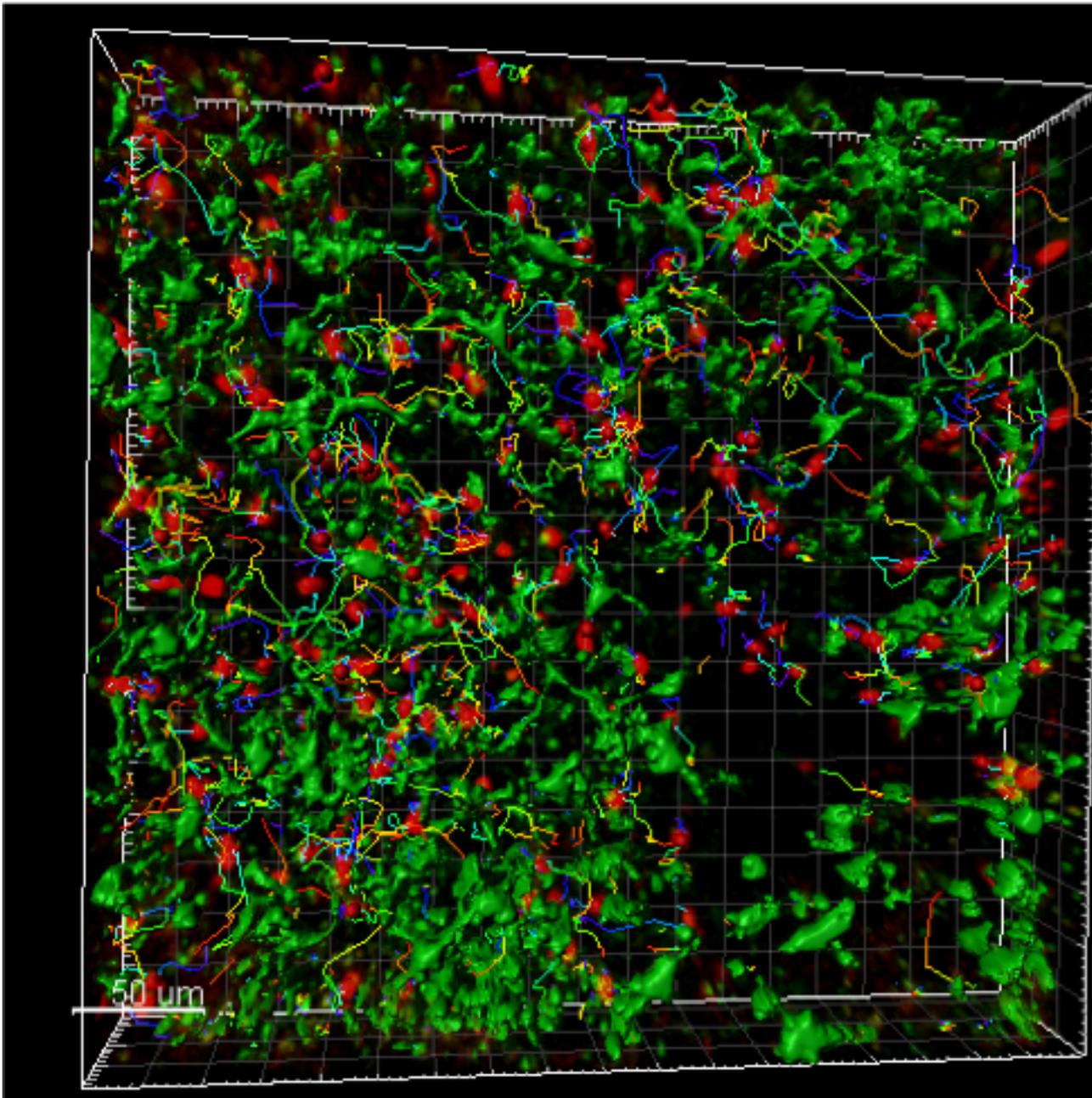
T cells move at an average speed of $0.11 \mu\text{m/s}$.

T cells searching systematically (raster scan) would discover an antigen target in 6 days on average.

Simple random walkers (Brownian) have an expected 30% success rate after 3 days.[1]

T cells are able to find cognate antigen in 3-8 hours and give up after 12-24 hours.

<-- DCs in green, T cells in red

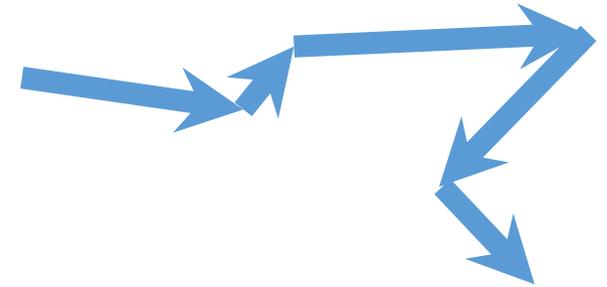


Janie Rae Byrum

[1] Preston, S. P., et al. "T-cell motility in the early stages of the immune response modeled as a random walk amongst targets." *Physical Review E* 74.1 (2006): 011910.

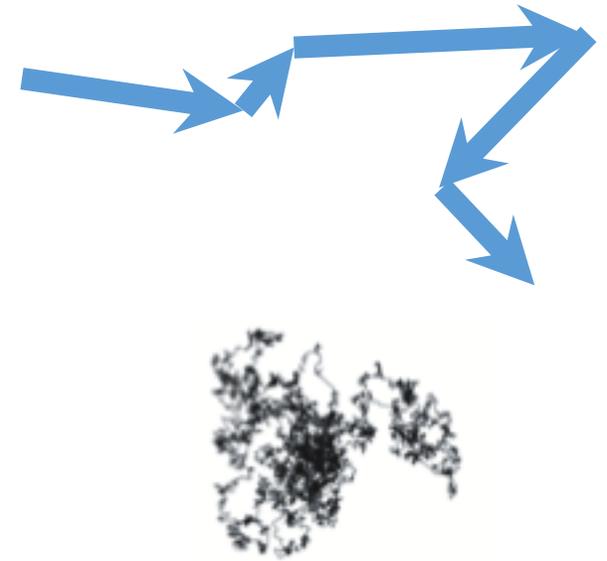
Background: Intensive vs Extensive Search

- We can describe any stochastic search pattern with distributions of vector lengths and turning angles.



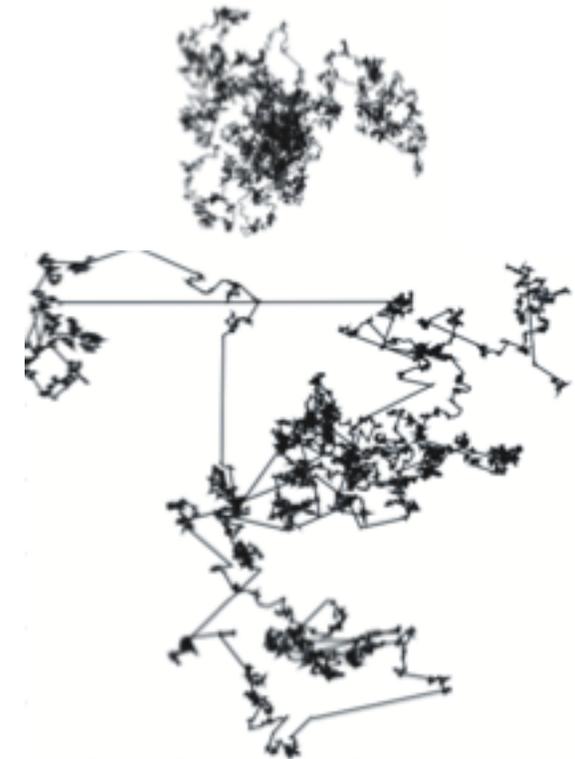
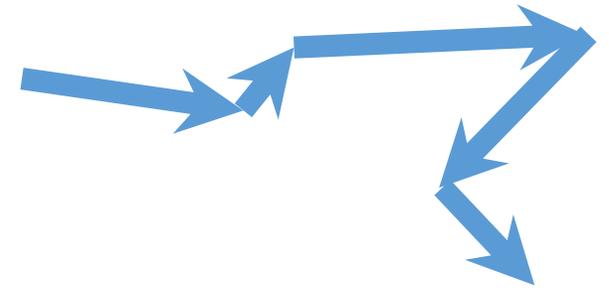
Background: Intensive vs Extensive Search

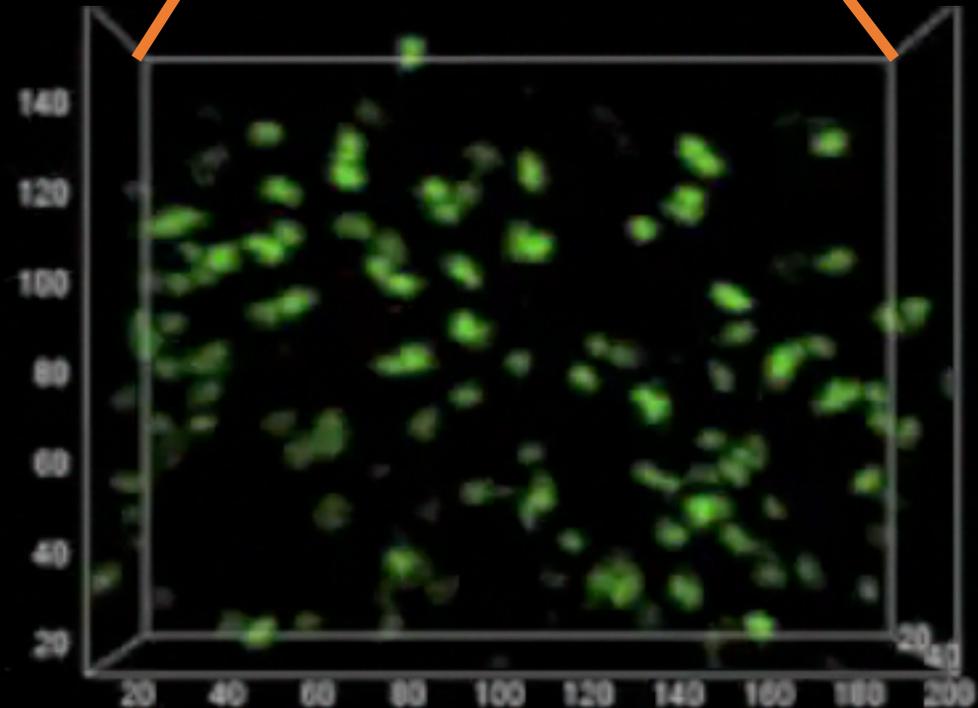
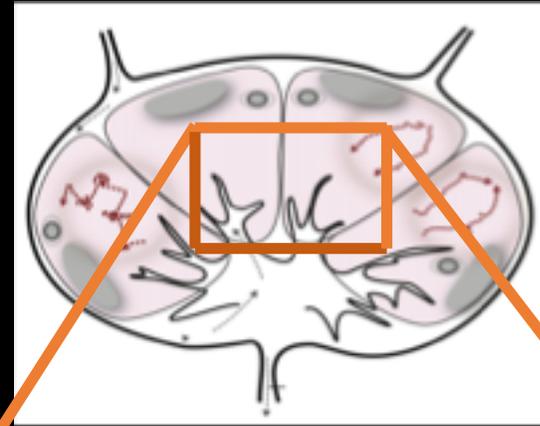
- We can describe any stochastic search pattern with distributions of vector lengths and turning angles.
- Intensive searchers have lower displacement but search more thoroughly.
- Simple random search (Brownian motion) will eventually cover the entire area.



Background: Intensive vs Extensive Search

- We can describe any stochastic search pattern with distributions of vector lengths and turning angles.
- *Intensive* searchers have lower displacement but search more thoroughly.
- Simple random search (Brownian motion) will eventually cover the entire area.
- *Extensive* searchers cover more ground but leave gaps.
- Mean Squared Displacement (MSD) is a measure of search extent



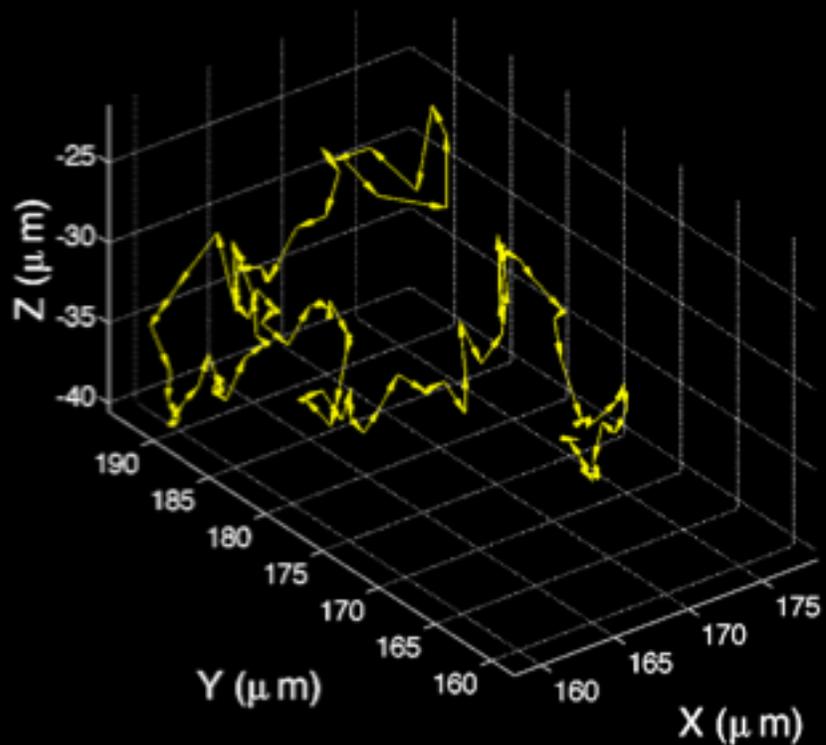
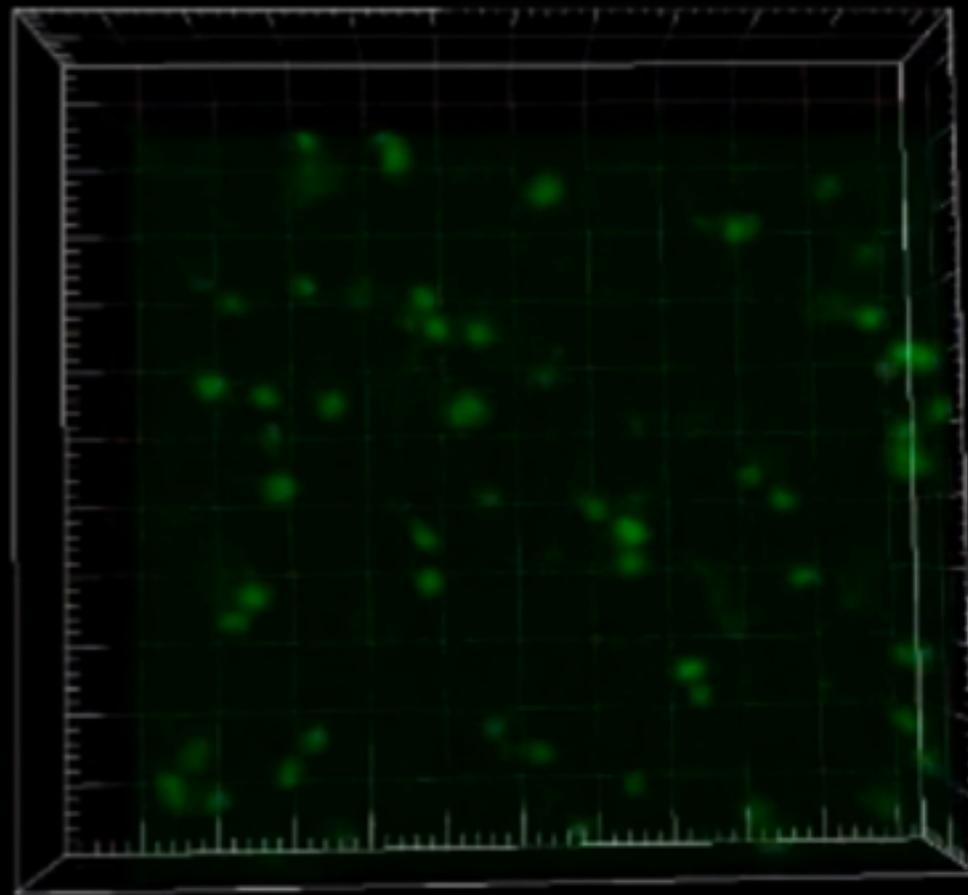
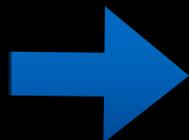
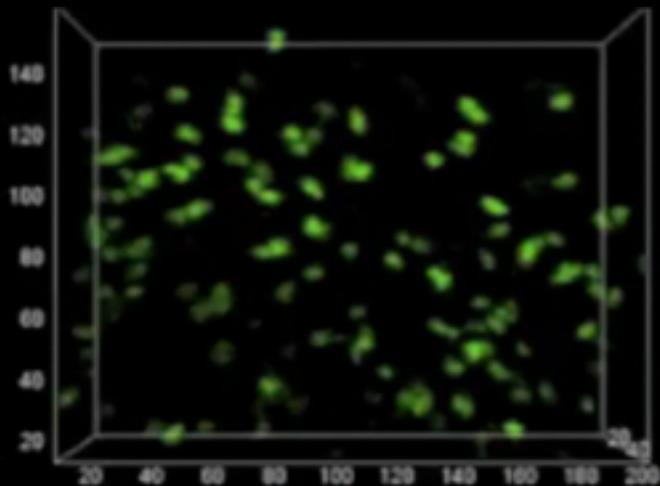


Two-photon microscopy

131 ex vivo observations

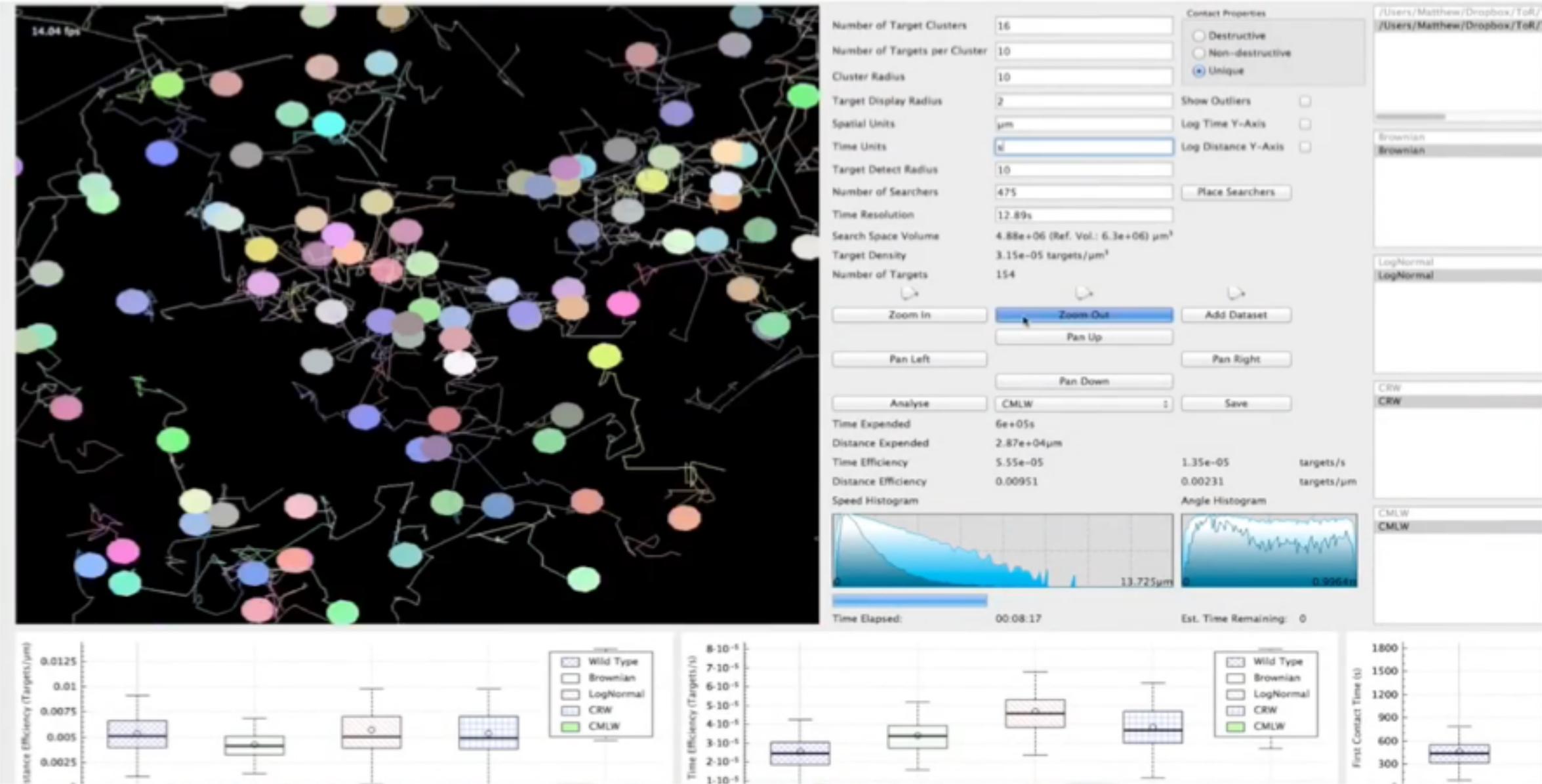
25,000 T cells tracked

Half hour observations



Extracting Tracks from Fluorescence
in collaboration with the Cannon lab

Building a statistical model of T cell Search



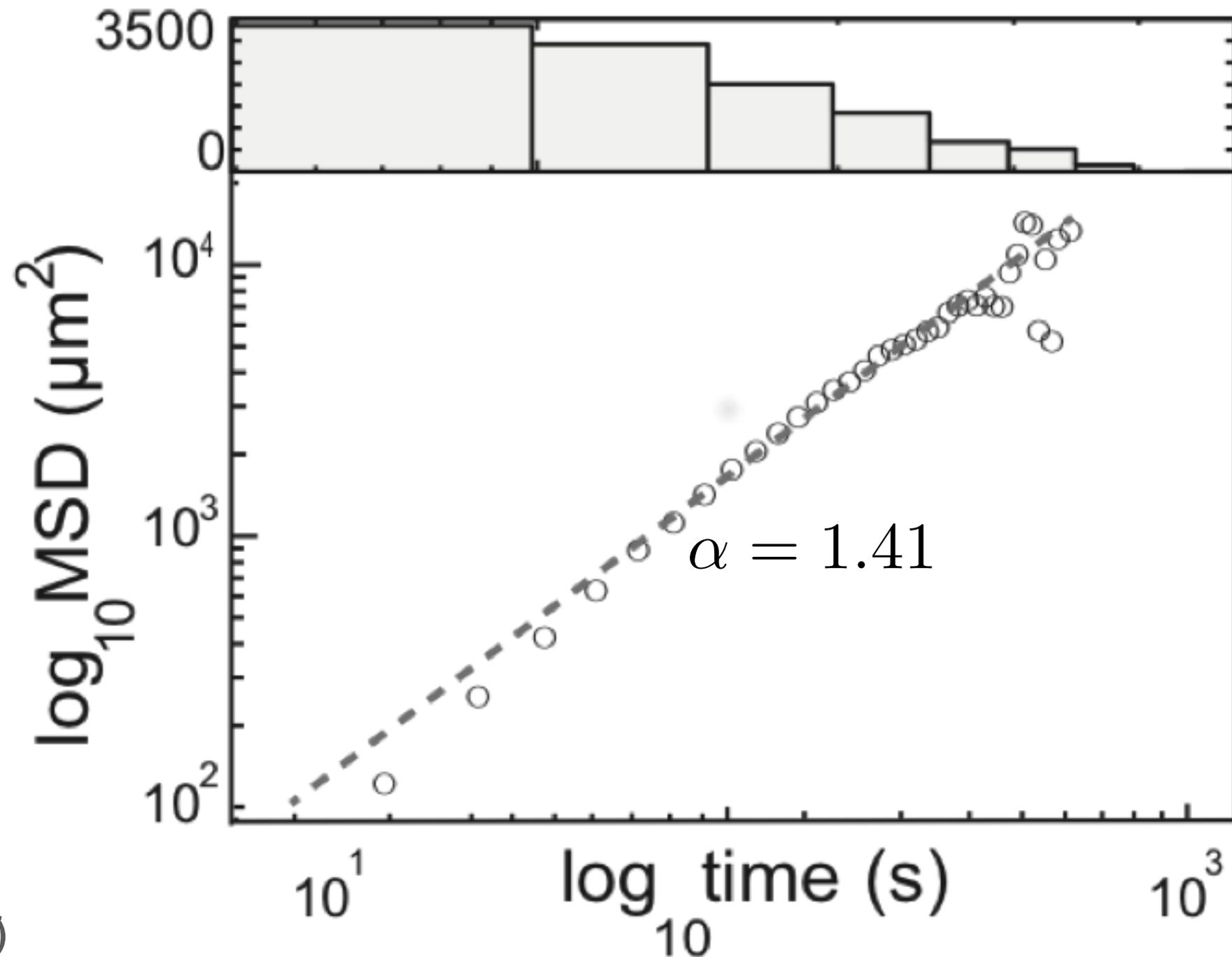
Displacement: Distance travelled from starting position.

Simple random walk
Would have a log transformed MSD with slope $\alpha = 1$ and $\mathcal{H} = 2$.

We see a slope of $\alpha = 1.41$
Superdiffusion.

Fractal dimension = 1.41

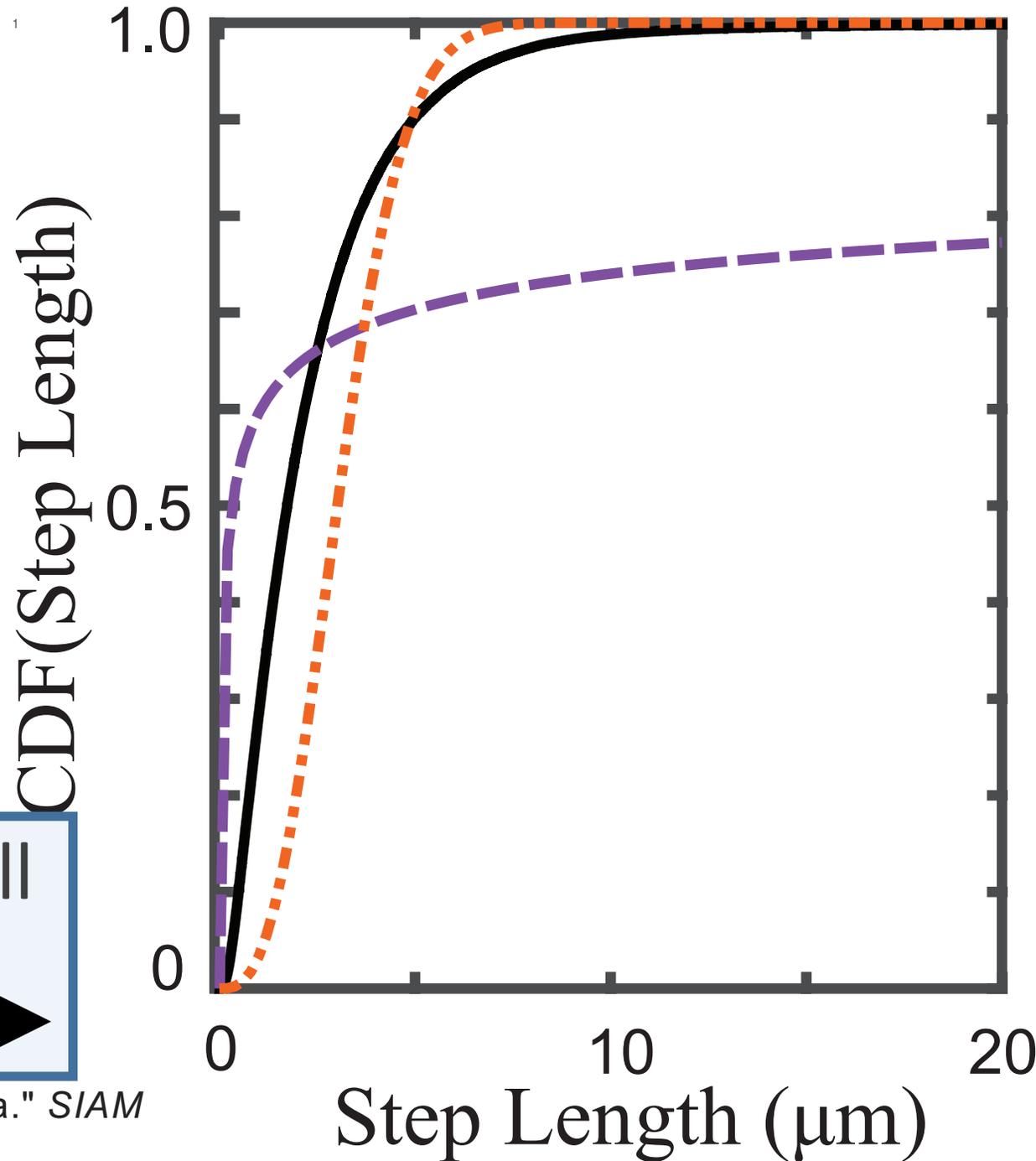
(This is our measure of the *intensity-extent trade-off*)



Modelling T cell Search: Fit Step Lengths



Maximum-likelihood fitting [1]



[1] Clauset, Aaron "Power-law distributions in empirical data." *SIAM review* (2009)

Modelling T cell Search: Fit Step Lengths

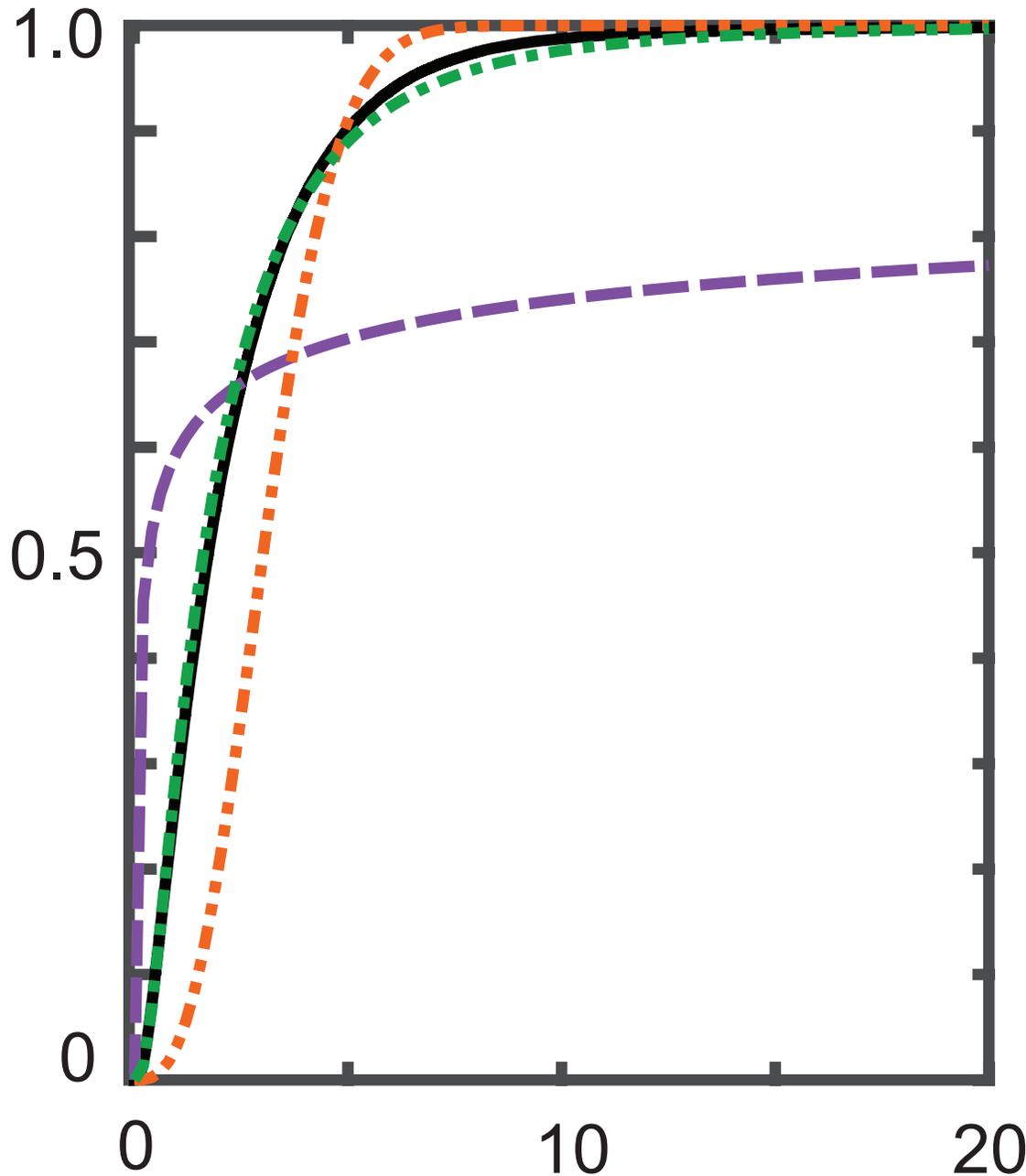
Observed 

Power Law 

Maxwell 

Lognormal 

CDF(Step Length)



Power Law Lognormal Maxwell

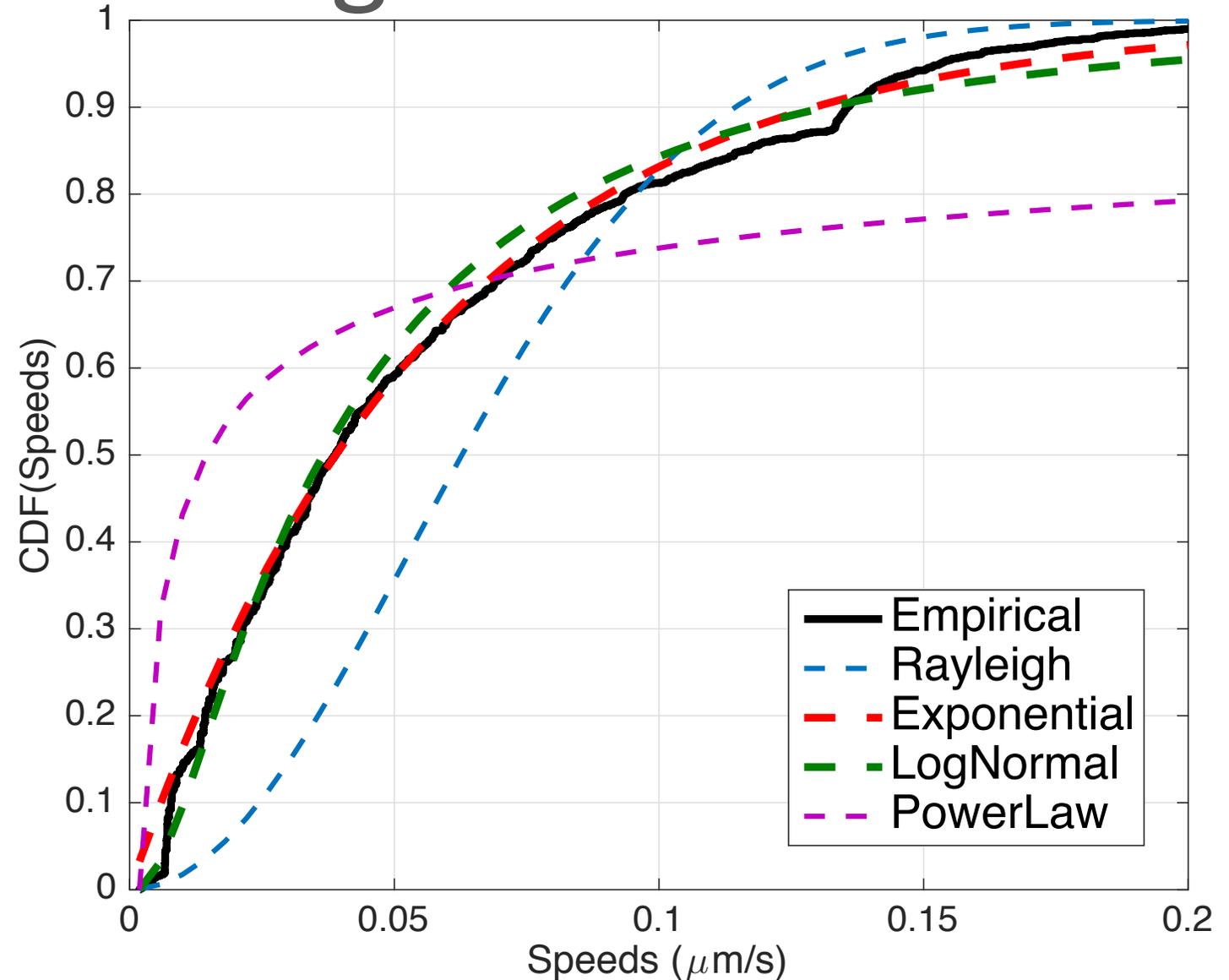
← Extensive

Intensive →

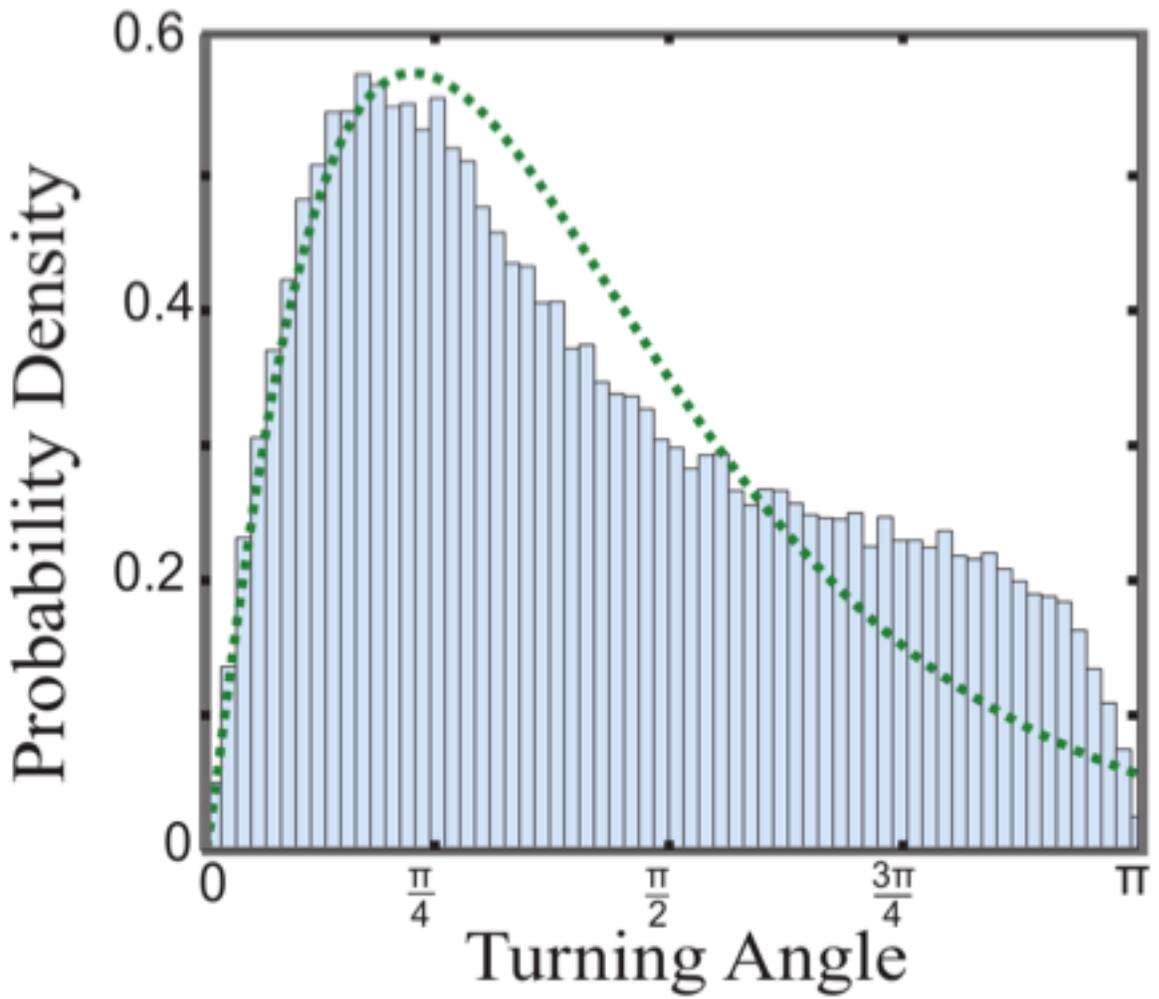
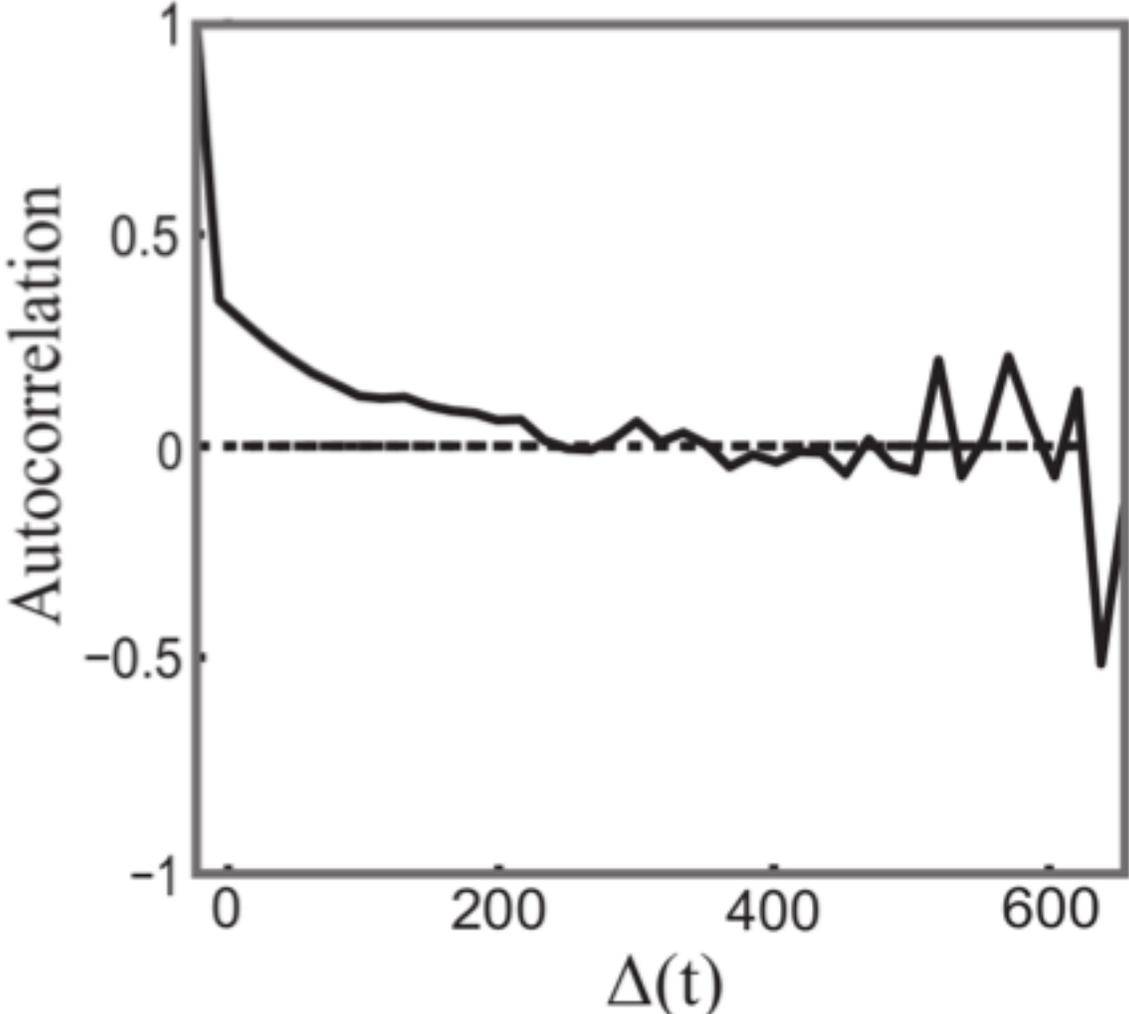
Step Length (µm)

T cell Search in the Lung

- The lognormal CDF is still a good fit.
- Exponential is also good.
- This pattern of movement is somewhat less intensive than in lymph nodes.



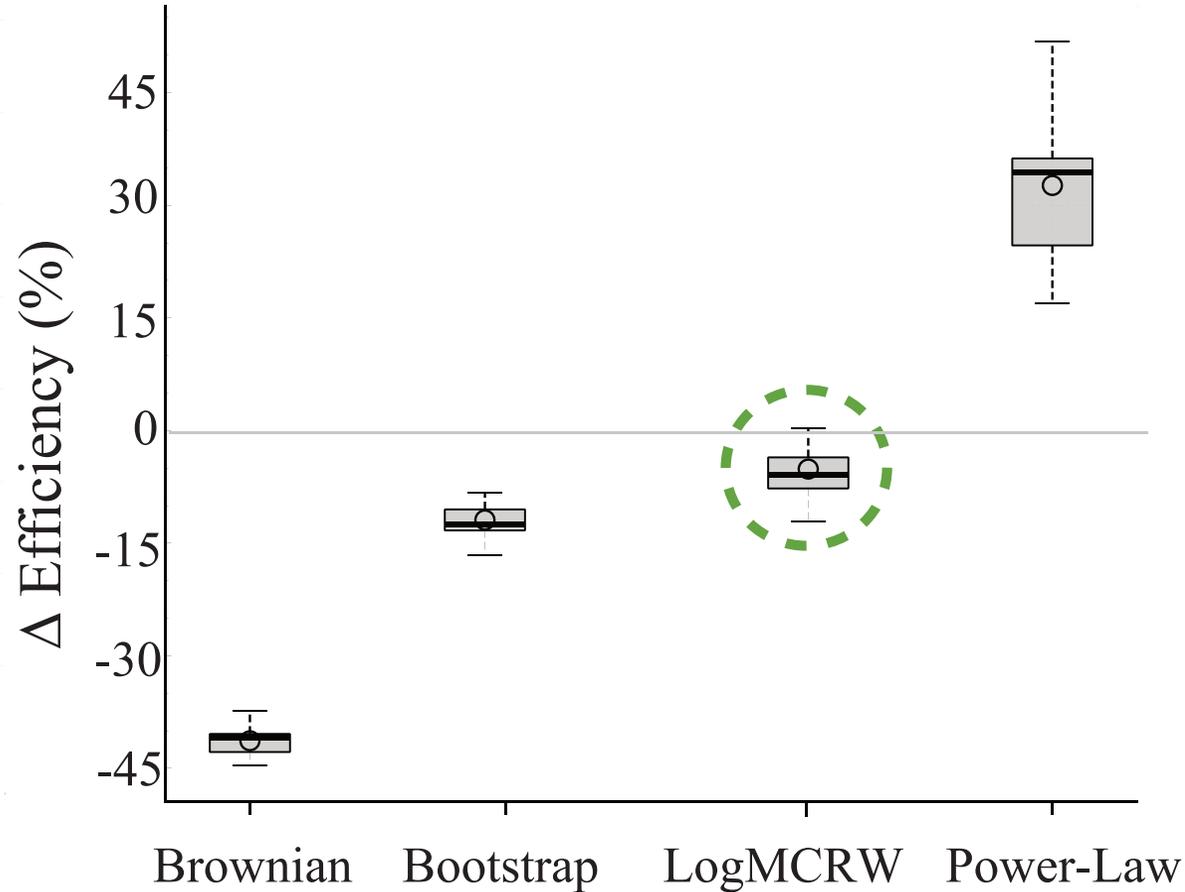
Modelling T cell Search: Angle Correlation (Search in Lungs and T cells look similar)



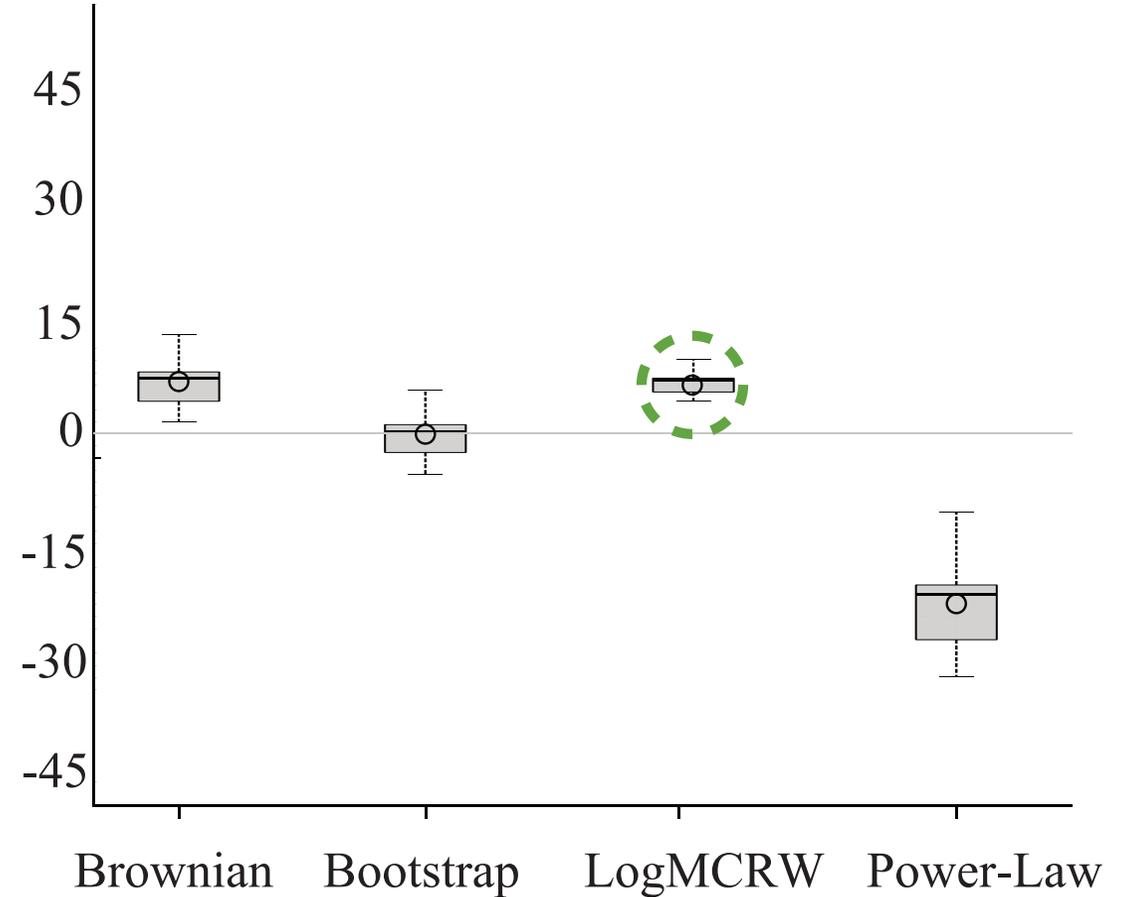
Measure of correlation with previous direction. Correlated random walk?

Comparison of search efficiency with empirical observations

Extensive search is better



Intense search is better



Unique Contacts

Total Contacts

Intensive

Extensive

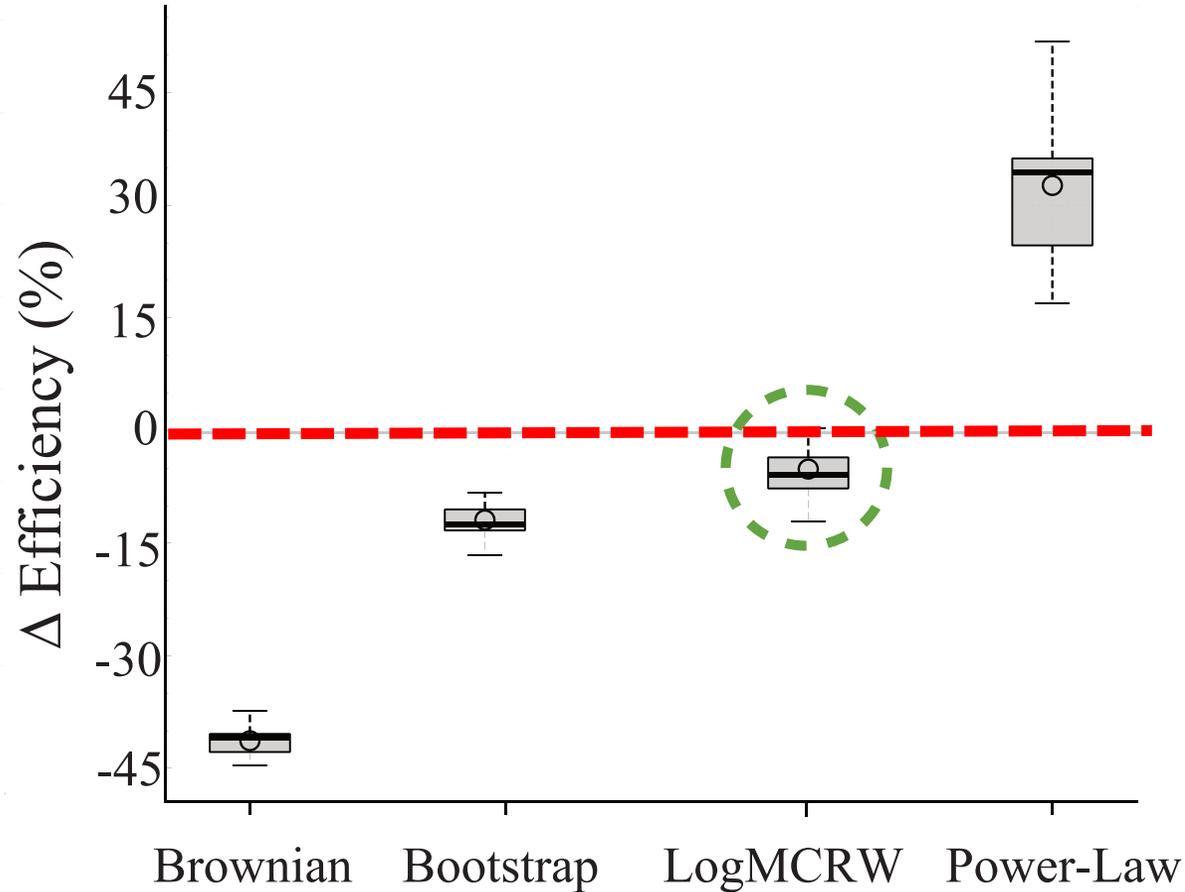
Intensive

Extensive

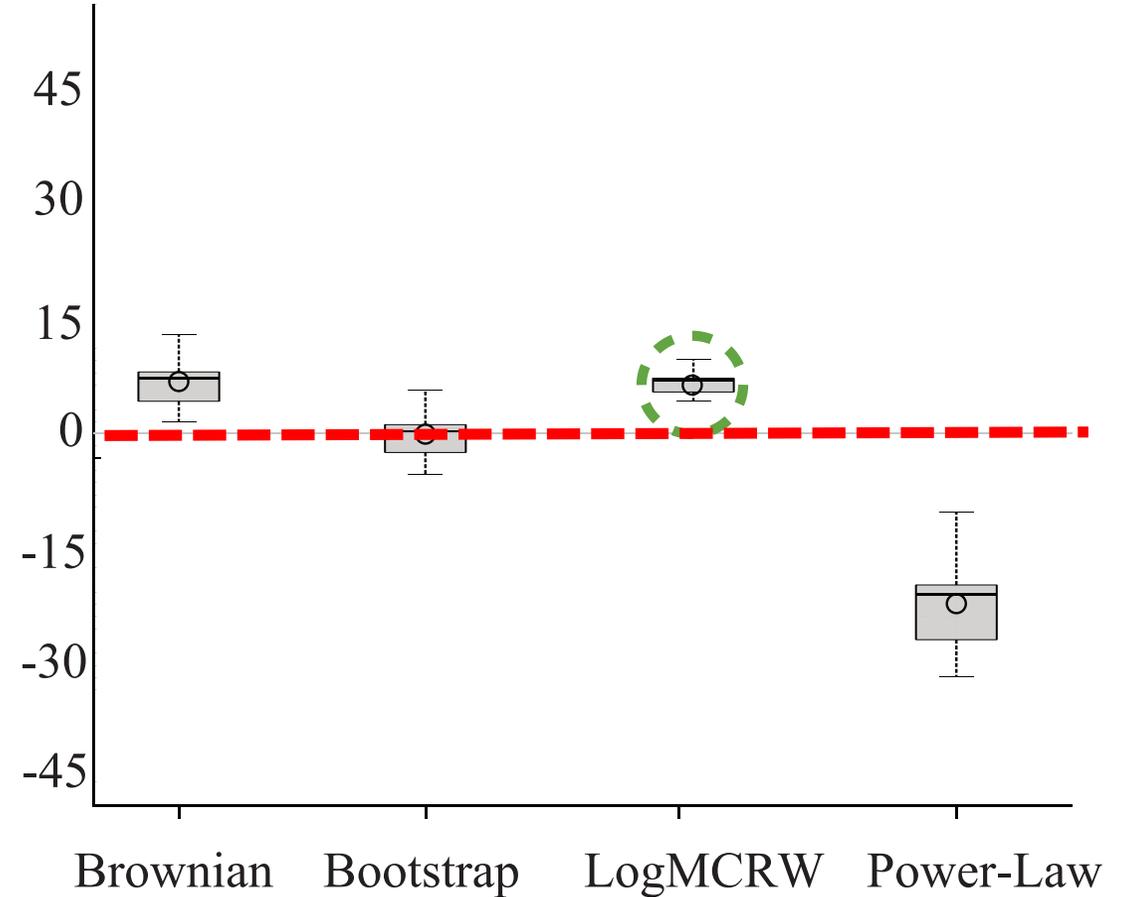
Intensity vs Extent Tradeoff: Target Detection Error

Comparison of search efficiency with empirical observations

Extensive search is better



Intense search is better



Unique Contacts

Total Contacts

Intensive

Extensive

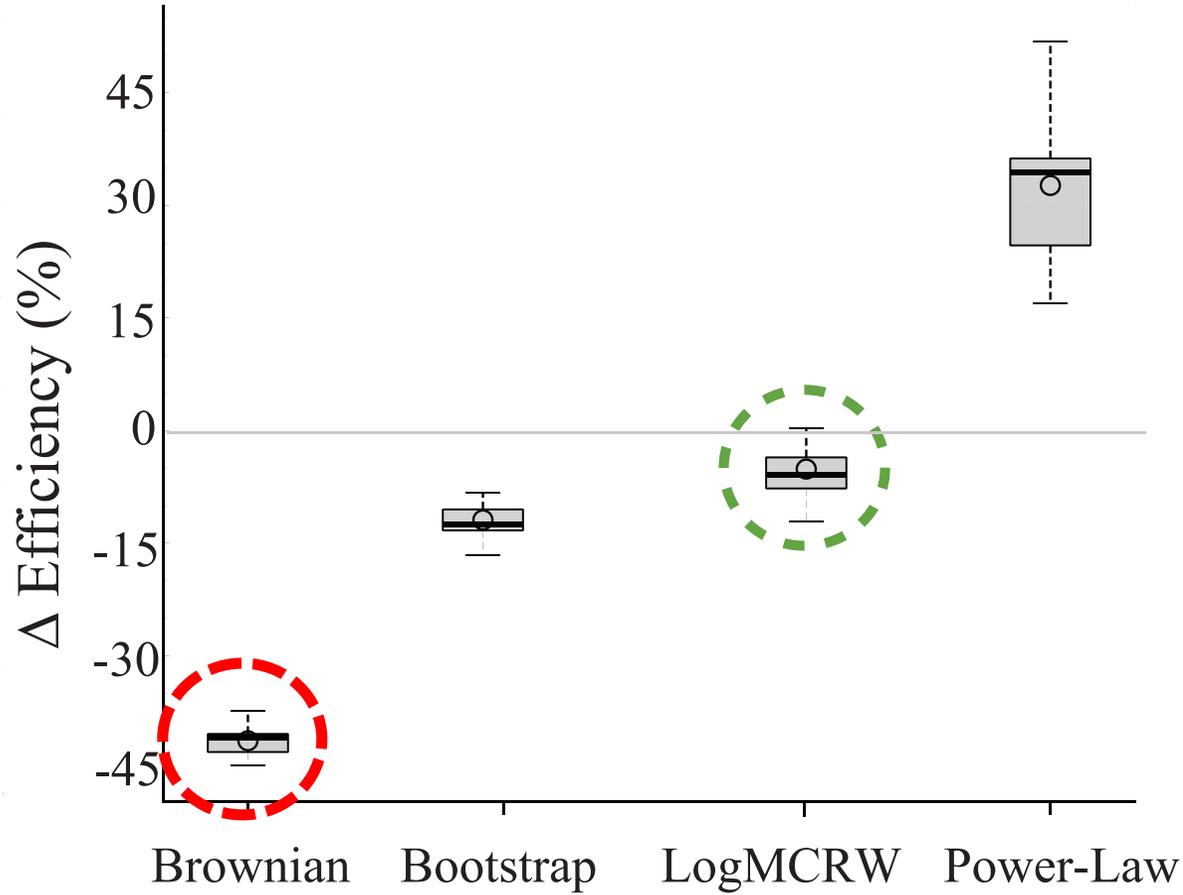
Intensive

Extensive

Intensity vs Extent Tradeoff: Target Detection Error

Comparison of search efficiency with empirical observations

Extensive search is better



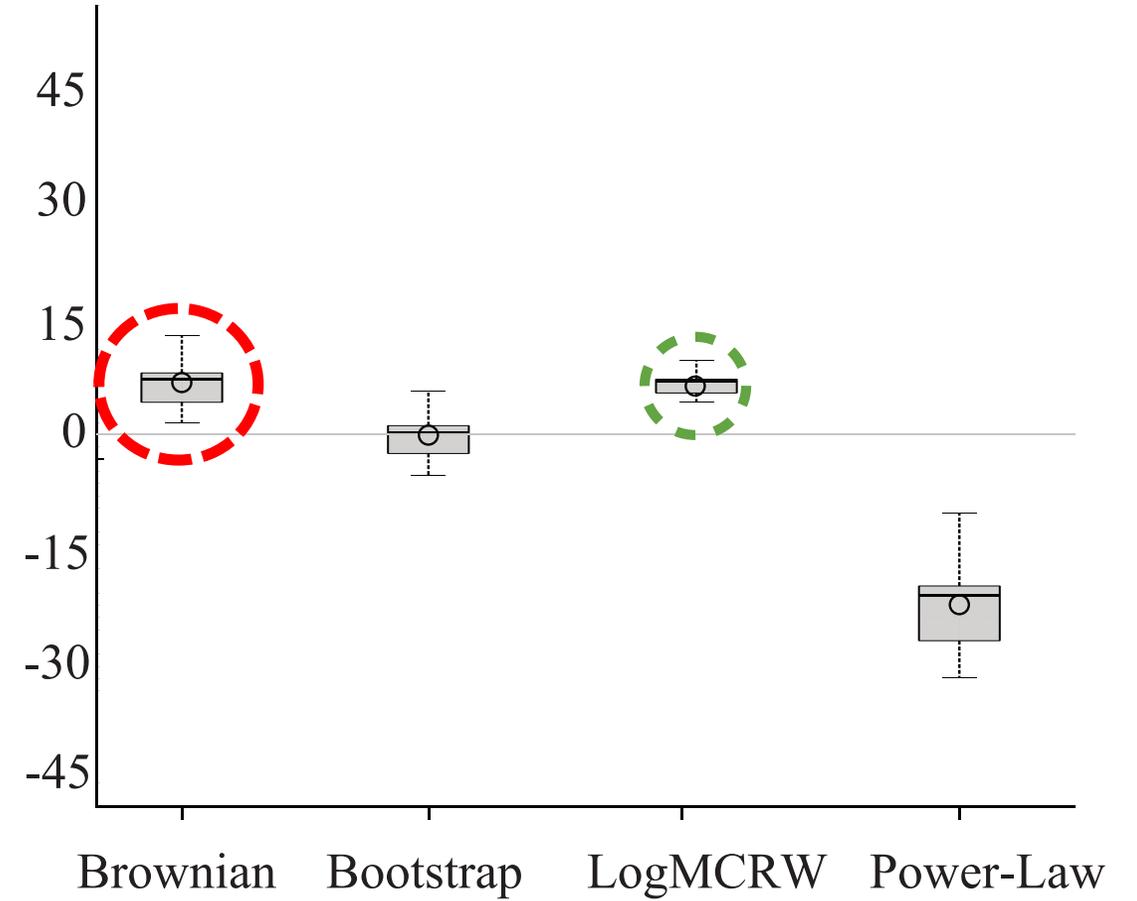
Unique Contacts

Intensive

Extensive

Intensity vs Extent Tradeoff: Target Detection Error

Intense search is better



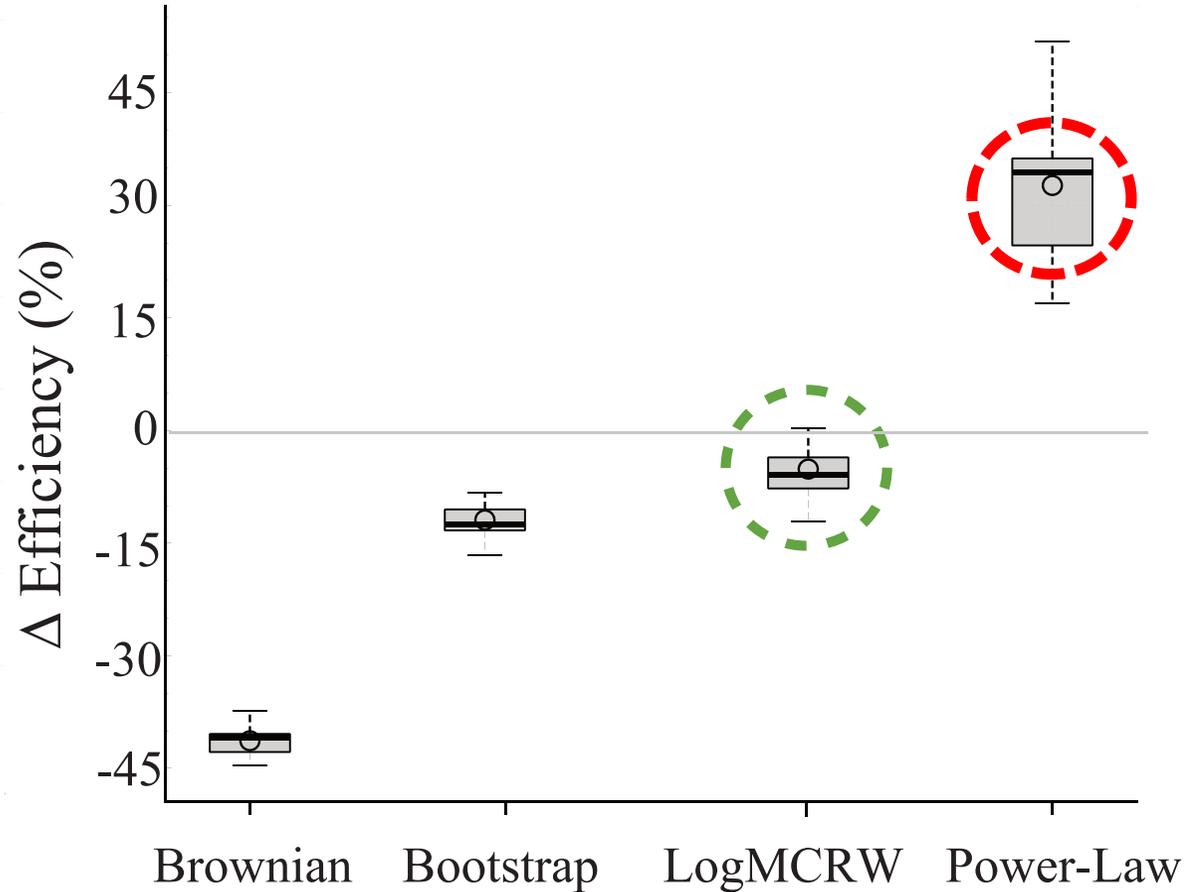
Total Contacts

Intensive

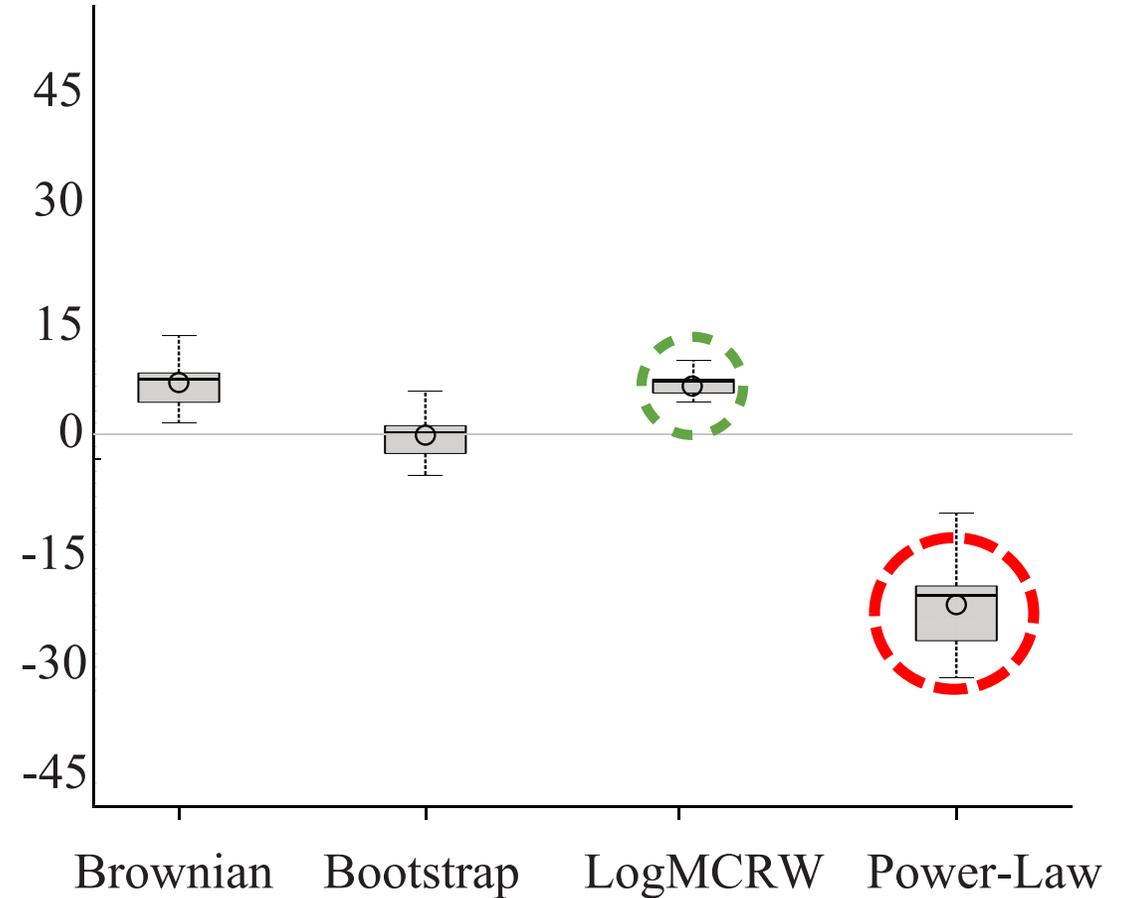
Extensive

Comparison of search efficiency with empirical observations

Extensive search is better



Intense search is better



Unique Contacts

Total Contacts

Intensive

Extensive

Intensive

Extensive

Intensity vs Extent Tradeoff: Target Detection Error

Why don't T cells use the parameters that result in the most unique contacts?

Why don't T cells use the parameters that result in the most unique contacts?

T cells have to revisit antigen multiple times for ligand and receptor rafts to form and signal to be properly integrated. [1,2]

[1] Fricke and Thomas, *BioPhysical Chemistry* (2006)

Why don't T cells use the parameters that result in the most unique contacts?

T cells have to revisit antigen multiple times for ligand and receptor rafts to form and signal to be properly integrated. [1,2]

As rarity (expected distance between targets and searcher) increases extensive search does better [3].

Intensive search does better when cognate antigen is common. T cells are able to take advantage of both.

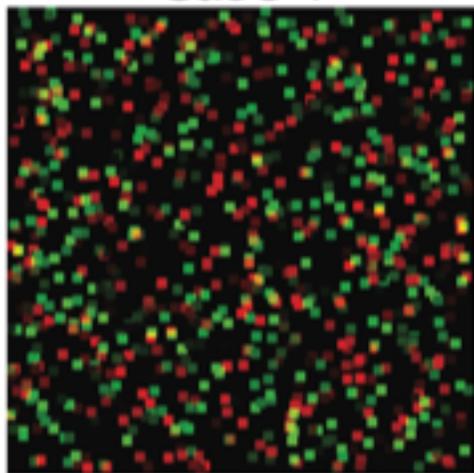
[1] Fricke and Thomas, *BioPhysical Chemistry* (2006)

[2] Celli, S. et al. *Immunity*, (2007)

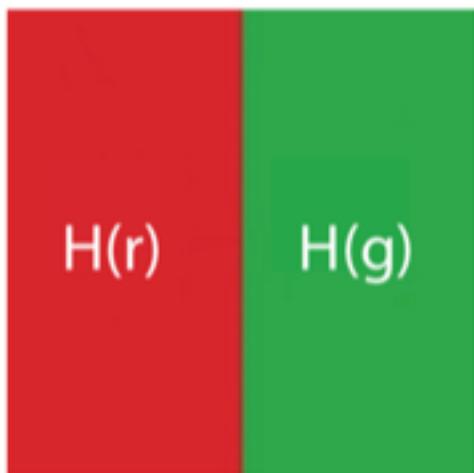
[3] Zhao, K., et al. *Journal of The Royal Society Interface* (2015)

Case 1

A

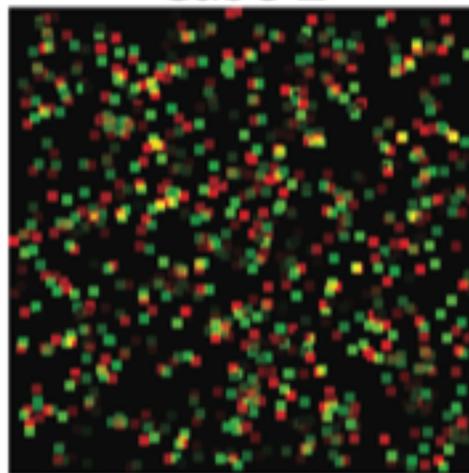


D

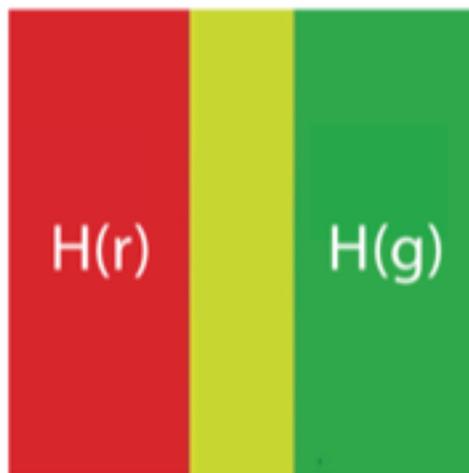


Case 2

B

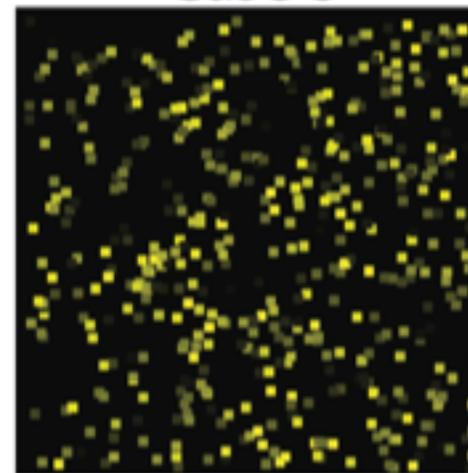


E

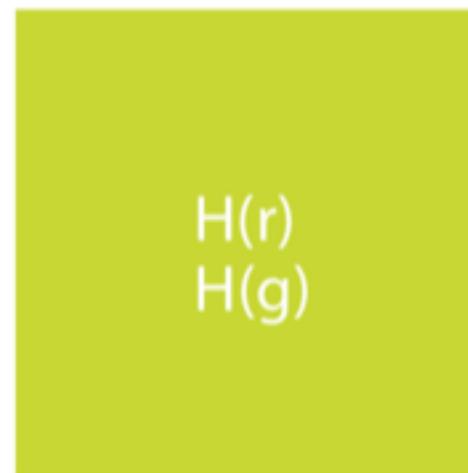


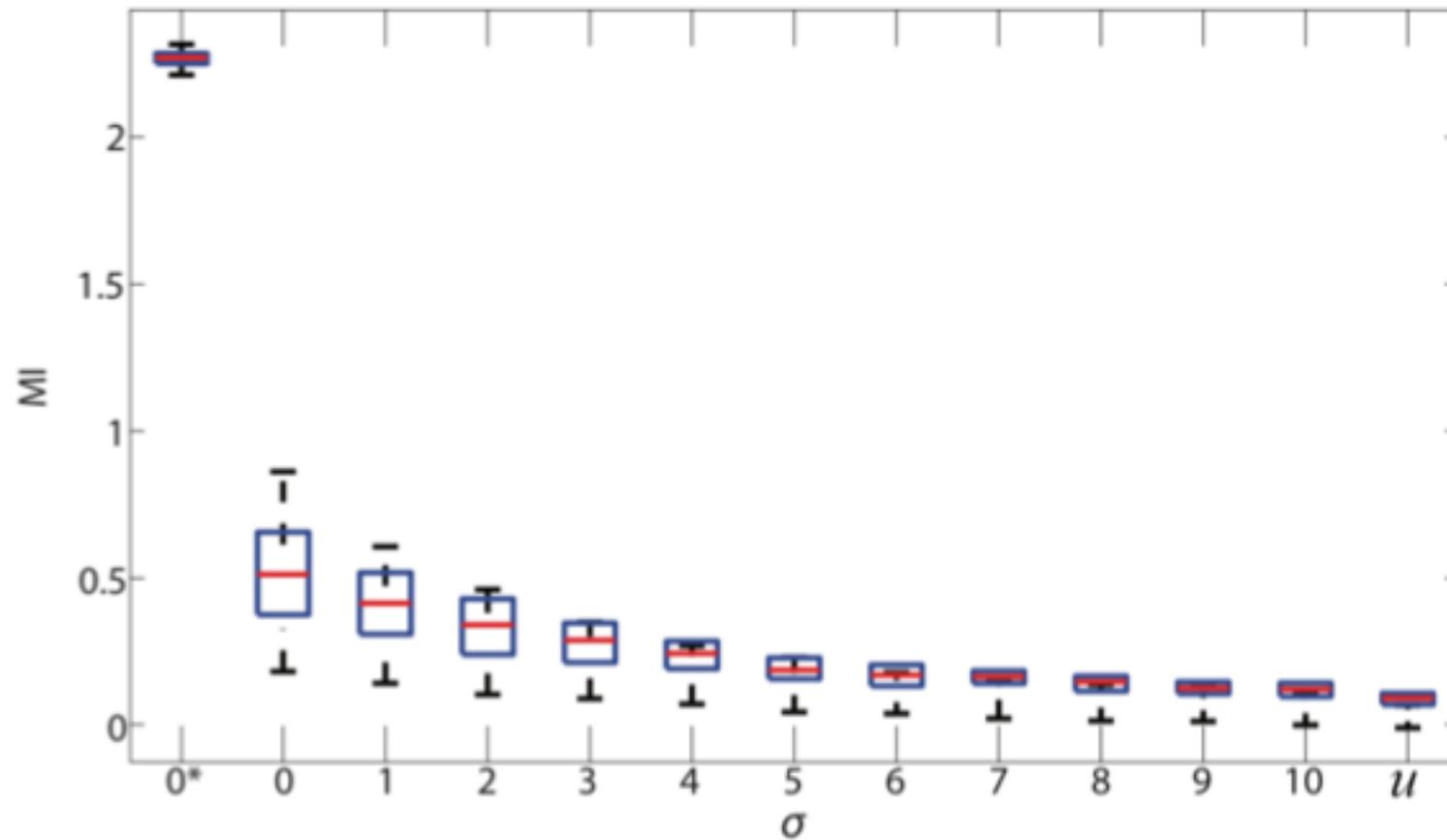
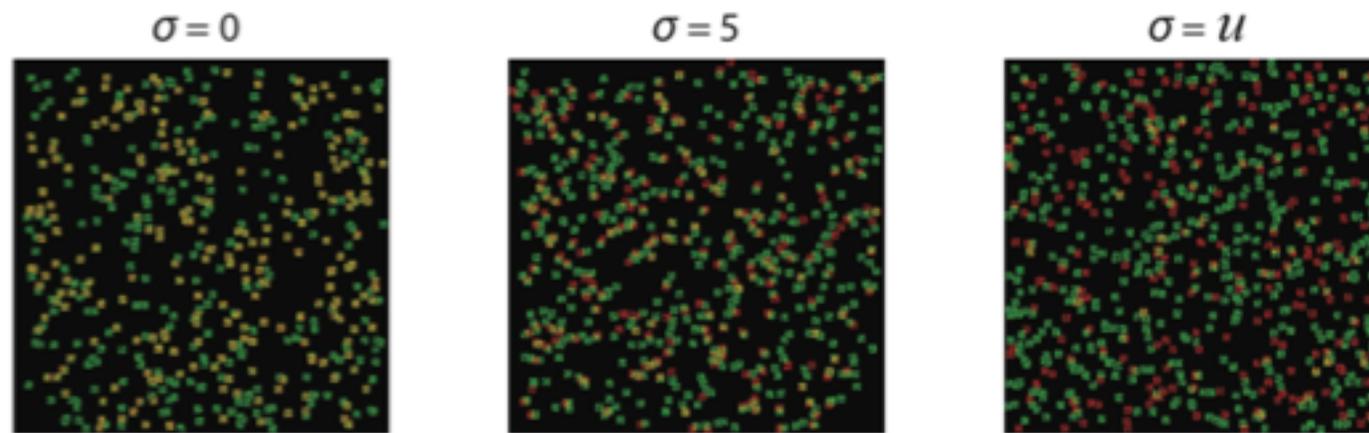
Case 3

C



F



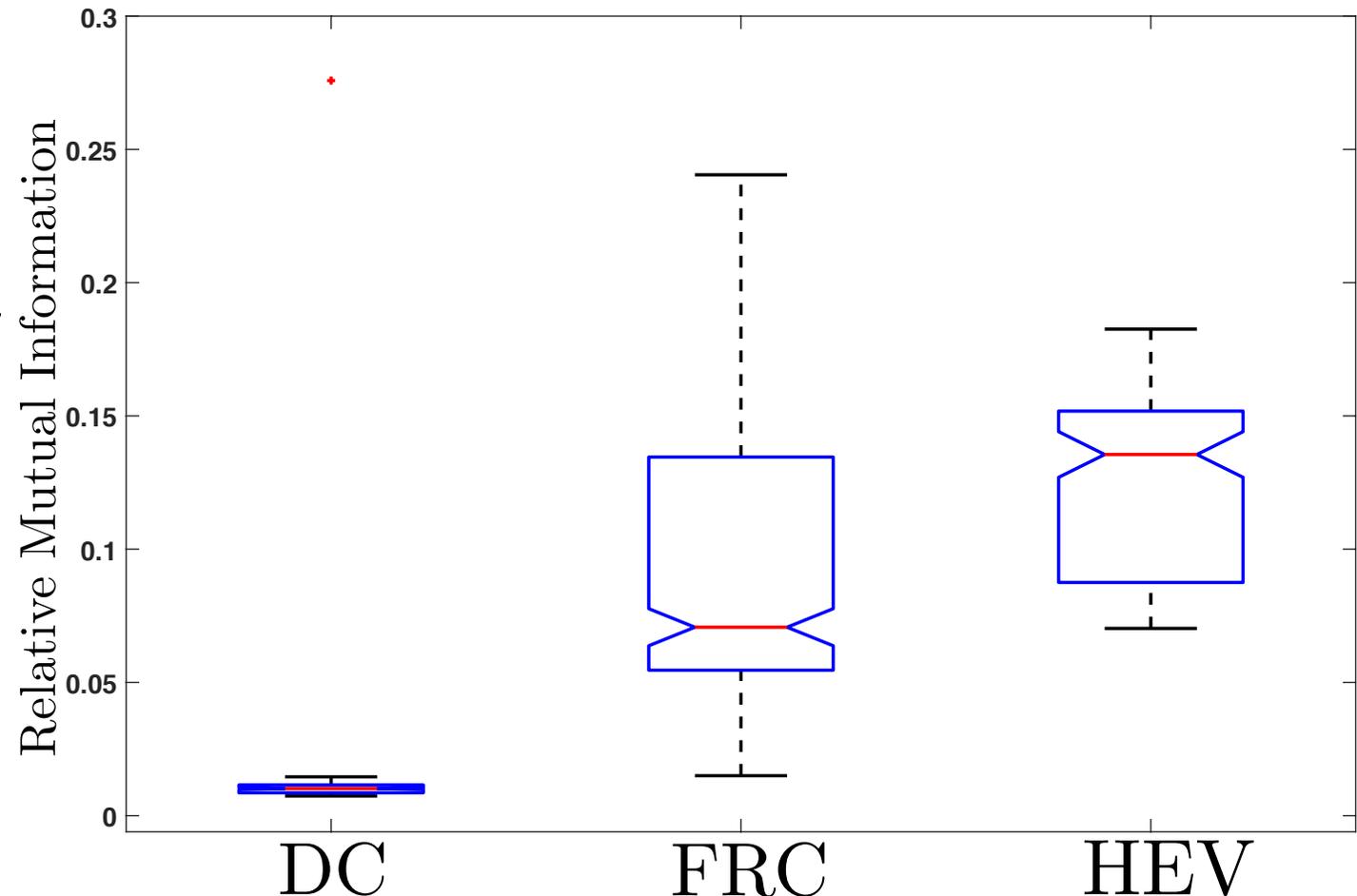
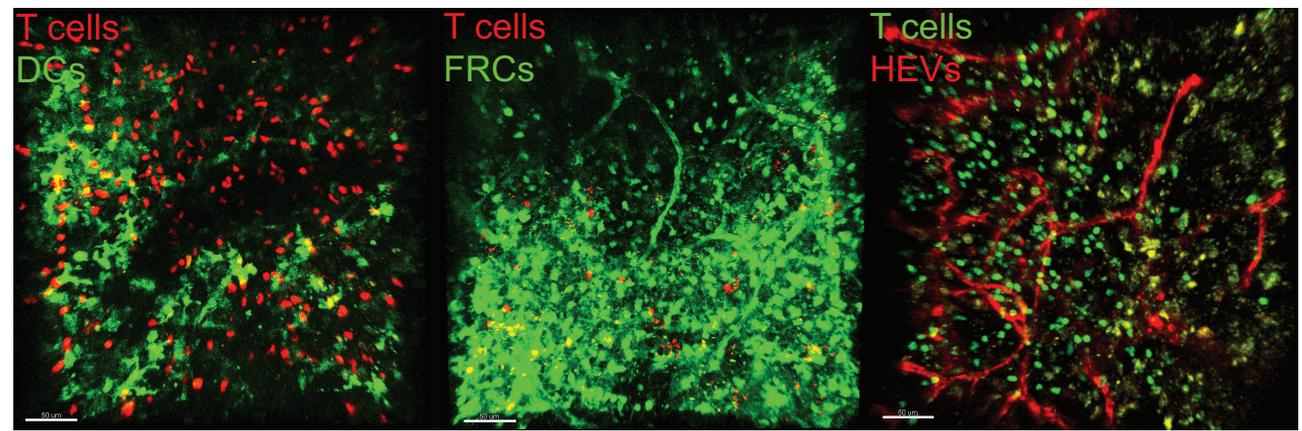


In other work: Beyond Random Walks

Fibroblastic Reticulum Cells (FRC)

There are theories that depend on T cell associations with FRC Network and HEVs [1,2,3]

No signs of chemical attraction Between T cells and DCs contrary to [4].



[1] Novkovic, Mario, et al. "Topological small-world organization of the fibroblastic reticular cell network determines lymph node functionality." *PLoS biology* 14.7 (2016): e1002515.

[2] Textor, Johannes, Judith N. Mandl, and Rob J. de Boer. "The reticular cell network: a robust backbone for immune responses." *PLoS biology* 14.10 (2016): e2000827.

[3] Girard, Jean-Philippe, Christine Moussion, and Reinhold Förster. "HEVs, lymphatics and homeostatic immune cell trafficking in lymph nodes." *Nature Reviews Immunology* 12.11 (2012): 762-773.

[4] Riggs, Thomas, et al. "A comparison of random vs. chemotaxis-driven contacts of T cells with dendritic cells during repertoire scanning." *Journal of theoretical biology* 250.4 (2008): 732-751.

T cell Search Summary

- T cells are superdiffusive when searching lymph nodes and lung tissue, helping to explain their efficiency.

T cell Search Summary

- T cells are superdiffusive when searching lymph nodes and lung tissue, helping to explain their efficiency.
- The search pattern in lymph nodes allows for signal integration (less important in lungs).

T cell Search Summary

- T cells are superdiffusive when searching lymph nodes and lung tissue, helping to explain their efficiency.
- The search pattern in lymph nodes allows for signal integration (less important in lungs).
- Associated with the FRC small world network which may also increase efficiency.

T cell Search Summary

- T cells are superdiffusive when searching lymph nodes and lung tissue, helping to explain their efficiency.
- The search pattern in lymph nodes allows for signal integration (less important in lungs).
- Associated with the FRC small world network which may also increase efficiency.
- Little spatial correlation with DCs which argues against DC recruitment.